

REPORT DOCUMENTATION PAGE			Form Approved
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the reviewing of existing data needed, and completing and reviewing the collection of information. Send comments to Washington Headquarters Services, Directorate for Information Operations and Reports, 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.			AFRL-SR-BL-TR-98- <i>0811</i>
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE December, 1995	3. REPORT TYPE AND DATES COVERED Final	
4. TITLE AND SUBTITLE USAF Summer Research Program - 1995 Summer Faculty Research Program Final Reports, Volume 6A, AEDC, FJSRL, WHMC, and ALCs			5. FUNDING NUMBERS
6. AUTHORS Gary Moore			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Research and Development Labs, Culver City, CA			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NI 4040 Fairfax Dr, Suite 500 Arlington, VA 22203-1613			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES Contract Number: F49620-93-C-0063			
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release			12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) The United States Air Force Summer Faculty Research Program (USAF- SFRP) is designed to introduce university, college, and technical institute faculty members to Air Force research. This is accomplished by the faculty members being selected on a nationally advertised competitive basis during the summer intersession period to perform research at Air Force Research Laboratory Technical Directorates and Air Force Air Logistics Centers. Each participant provided a report of their research, and these reports are consolidated into this annual report.			
14. SUBJECT TERMS AIR FORCE RESEARCH, AIR FORCE, ENGINEERING, LABORATORIES, REPORTS, SUMMER, UNIVERSITIES			15. NUMBER OF PAGES
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

UNITED STATES AIR FORCE
SUMMER RESEARCH PROGRAM -- 1995
SUMMER FACULTY RESEARCH PROGRAM FINAL REPORTS

VOLUME 6A

ARNOLD ENGINEERING DEVELOPMENT CENTER
FRANK J. SEILER RESEARCH LABORATORY
WILFORD HALL MEDICAL CENTER
AIR LOGISTIC CENTERS

RESEARCH & DEVELOPMENT LABORATORIES
5800 Uplander Way
Culver City, CA 90230-6608

Program Director, RDL
Gary Moore

Program Manager, AFOSR
Major David Hart

Program Manager, RDL
Scott Licoscas

Program Administrator, RDL
Gwendolyn Smith

Program Administrator, RDL
Johnetta Thompson

Submitted to:

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Bolling Air Force Base

Washington, D.C.

December 1995

19981214 079

DTIC QUALITY INSPECTED 3

PREFACE

Reports in this volume are numbered consecutively beginning with number 1. Each report is paginated with the report number followed by consecutive page numbers, e.g., 1-1, 1-2, 1-3; 2-1, 2-2, 2-3.

Due to its length, Volume 6 is bound in two parts, 6A and 6B. Volume 6A contains #1-12, and Volume 6B contains reports #13-23. The Table of Contents for Volume 6 is included in both parts.

This document is one of a set of 16 volumes describing the 1995 AFOSR Summer Research Program. The following volumes comprise the set:

<u>VOLUME</u>	<u>TITLE</u>
1.	Program Management Report
	<i>Summer Faculty Research Program (SFRP) Reports</i>
2A & 2B	Armstrong Laboratory
3A & 3B	Phillips Laboratory
4	Rome Laboratory
5A, 5B, & 5C	Wright Laboratory
6A & 6B	Arnold Engineering Development Center, Wilford Hall Medical Center and Air Logistics Centers
	<i>Graduate Student Research Program (GSRP) Reports</i>
7A & 7B	Armstrong Laboratory
8	Phillips Laboratory
9	Rome Laboratory
10A & 10B	Wright Laboratory
11	Arnold Engineering Development Center, Wilford Hall Medical Center and Air Logistics Centers
	<i>High School Apprenticeship Program (HSAP) Reports</i>
12A & 12B	Armstrong Laboratory
13	Phillips Laboratory
14	Rome Laboratory
15A&15B	Wright Laboratory
16	Arnold Engineering Development Center

SFRP FINAL REPORT TABLE OF CONTENTS

i-xiv

1. INTRODUCTION	1
2. PARTICIPATION IN THE SUMMER RESEARCH PROGRAM	2
3. RECRUITING AND SELECTION	3
4. SITE VISITS	4
5. HBCU/MI PARTICIPATION	4
6. SRP FUNDING SOURCES	5
7. COMPENSATION FOR PARTICIPATIONS	5
8. CONTENTS OF THE 1995 REPORT	6

APPENDICIES:

A. PROGRAM STATISTICAL SUMMARY	A-1
B. SRP EVALUATION RESPONSES	B-1

SFRP FINAL REPORTS

Associate did not participate in program.

ANALYSIS OF TURBULENT, SUPERSONIC, AND AXISYMMETRIC PLUME DATA

Saad A. Ahmed

Associate Professor

Mechanical Engineering Department

KFUPM, Dhahran 31261, KSA

Final report for
Summer Faculty Research Program
Arnold Engineering and Development Center

Sponsored by:

Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC

and

Arnold Engineering and Development Center

September, 1995

ANALYSIS OF TURBULENT, SUPERSONIC, AND AXISYMMETRIC PLUME DATA

Saad A. Ahmed
Associate Professor
Mechanical Engineering Department
KFUPM, Dhahran 31261, KSA

ABSTRACT

Analysis of experimental data obtained in the flowfield of a generic tactical missile configuration in simulated powered flight is performed. Mean velocities, Reynolds stresses, triple products, and higher order statistics measured by a two component LDV system working, in the off-axis forward scatter mode, were analyzed. A balance of the turbulence energy equation have been performed in order to get a more detailed insight into the turbulent behavior. The analysis utilized the experimental LDV data to determine the turbulence production, diffusion, convection and viscous dissipation. The analysis of the data was successful in identifying the various areas of interest in the flowfield where different turbulent transport mechanisms dominate.

ANALYSIS OF TURBULENT, SUPERSONIC, AND AXISYMMETRIC PLUME DATA

Saad A. Ahmed

1 INTRODUCTION

The flowfield in the vicinity of the base of a missile has been a subject of interest in the past few years. The adverse effects of base drag (a major component of the total drag) on tactical missile cruise performance is well known. Plume induced separation of missile afterbody flow would cause severe degradation of the stability characteristics of a vehicle. Also, the effectiveness of control surfaces could be reduced (i.e., due to separation of flow in this region) and catastrophic failure might result. The complexity and significance of the flowfield in this region make missile base-flow accurate predictions difficult and essential. Although substantial literature exists (see Dutton et al. 1995 for more details), it is neither complete nor sufficient to match the progress that has been made in computational fluid dynamics capabilities and computers in recent years. This lack of detailed experimental data has been a major restriction to the development and validation of numerical prediction codes that model the flowfield of missile base flow. High quality and detailed experimental data utilizing non-intrusive diagnostic methods are needed for development and validation of both new and improved or modified current computational methods (see Kennedy et al. 1992).

2 EXPERIMENTAL SET-UP

The experiments were conducted in AEDC Aerodynamic Wind Tunnel (1 T) which is a continuous flow, and an open circuit wind tunnel. It can operate at a constant stagnation pressure of 1.35 atm over a Mach number range from 0.2 to 1.5. The stagnation temperature could be varied from 300 K to 323 K above ambient temperature. The floor and ceiling portions of the test sections were perforated in order to reduce wall interference at the test

section. The side walls were fitted with large flat and polished optical quality glass windows. Compressed air for the model jet plume was obtained from the VKF which could provide dry air, continuously at 200 atm. The air was supplied to the base nozzle through passages machined in the mounting strut. The tunnel was operated at a free stream Mach No. of 1.4 with a stagnation pressure of 137.5 kPa and a stagnation temperature of 355.6 K, or 640°R. The propulsive jet was operated at a pressure of 2.41×10^6 Pa (or 350 psia, NPR = 330, total nozzle exit pressure divided by static free stream pressure) and a Mach Number of 2.7. The high operating pressure resulted in highly under-expanded plume and a region of plume induced flow separation could develop around the base. The model was 57 cm long, 1.27 cm diameter ($L/D = 45$), circular ogive cylinder mounted on struts from the side walls (see Fig. 1). The details of the model and the tunnel are reported in details by Walker and Helstley, 1993.

2.1 LDV System

The laser Doppler velocimeter (LDV) system used was a two-component/ two-color (green and blue) operating in the off-axis forward scatter mode. The beams were frequency shifted using Bragg cells to determine the direction of the flow easily. The laser source was a 5-Watt argon ion laser, water cooled type. Fringes were oriented at $\pm 45^\circ$ relative to the tunnel axis to minimize fringe bias effects and to improve the resolution of the system (see Ahmed et al., 1992). The sampling volume was approximately 0.3 mm in diameter and 2.0 mm long. Probe volume location was expressed in x, y, z coordinates relative to an origin at the model nozzle exit center. Seed particles were injected both into the free stream flow and into the high pressure air jet utilizing liquid atomizers or fluidized bed. The details of the system and its operational procedure were reported by Walker and Helstley, 1993.

3 METHODOLOGY

3.1 Reynolds' Equations:

A century ago, Reynolds (1895) published his famous paper which defined Reynolds stresses (see Moore et al., 1994). He considered the instantaneous value of any variable (i.e., velocity,

density, and pressure) to be composed of a mean and a fluctuating component about that mean, see Appendix A. After he substituted with the instantaneous values into the Navier - Stokes equations, he performed Reynolds averaging of the equations and showed that the new time - averaged Navier - Stokes equations contained additional new unknown terms. The new terms are products of fluctuating velocities named after him Reynolds stresses (i.e., $-\rho\overline{u'v'}$, $-\rho\overline{u'w'}$, $-\rho\overline{v'w'}$, etc.); see Appendix A. With the introduction of these new terms in the Navier-Stokes equations (Reynolds time-averaged equations), the number of equations for a turbulent flow is less than the number of unknowns. Therefore, a closure or turbulence model is needed in order to solve the equations of a turbulent flowfield. A simple model could relate the unknown Reynolds stresses to some mean flow quantities. This method is widely used nowadays for solving most of the practical engineering turbulent flow problems. The degree of complexity/simplicity of the model and its accurate prediction of the flow was and will be the subject of research for many years.

In practice, most of turbulence - model- development and applications were carried out in the area of mechanical and aeronautical engineering. In most flows of practical interest, turbulent motions characterized by large irregular spatial and temporal fluctuations contribute significantly to the transport of momentum, heat and mass. They have great influences on the distributions of velocity, temperature, pressure and species concentrations over the turbulent flowfield (see Hinze, 1975; Tennekes and Lumley 1972; Harlow and Nakayama, 1968; and Launder et al., 1975). It is the task of researchers working in the field of computational fluid mechanics to determine these distributions for a certain engineering flow problem of interest. If the problem is to be solved numerically, there is no way around making assumptions about the turbulent transport processes. This is what is meant by turbulence modeling. In simple words, the turbulent transport processes can not be calculated with an exact method but rather, they must be approximated by a turbulence model. This model could allow directly or indirectly, the unknown terms to be related to the mean flow properties.

3.2 Turbulence Energy Equation

The turbulence energy equation in cylindrical coordinates (for steady, axisymmetric, constant density flow, and $\overline{W} = 0.0$) could be written as (see Appendix B):

$$\begin{aligned}
\overline{U} \frac{\partial k}{\partial x} + \overline{V} \frac{\partial k}{\partial r} = & \frac{1}{\rho} \frac{\partial}{\partial x} \left[\mu \frac{\partial k}{\partial r} - \rho \frac{\overline{u'v'v'}}{2} - \rho \frac{\overline{u'w'w'}}{2} - \rho \frac{\overline{u'u'u'}}{2} - \overline{u'p'} \right] + \\
\frac{1}{\rho r} \frac{\partial}{\partial r} \left[\mu r \frac{\partial k}{\partial r} - \rho r \frac{\overline{v'v'v'}}{2} - \rho r \frac{\overline{v'w'w'}}{2} - \rho r \frac{\overline{u'u'v'}}{2} - r \overline{v'p'} \right] - & \left[\frac{2}{\rho} \mu \frac{\overline{v'v'}}{r^2} - \frac{\mu}{\rho} \frac{\overline{w'w'}}{r^2} \right] - \\
\left[\overline{v'v'} \frac{\partial \overline{V}}{\partial r} + \overline{u'v'} \frac{\partial \overline{U}}{\partial r} - \overline{u'v'} \frac{\partial \overline{V}}{\partial x} + \overline{u'u'} \frac{\partial \overline{V}}{\partial x} + \overline{w'w'} \frac{\overline{V}}{r} \right] - & \\
\frac{\mu}{\rho} \left[\left(\frac{\partial v'}{\partial r} \right)^2 + \left(\frac{\partial w'}{\partial r} \right)^2 + \left(\frac{\partial u'}{\partial r} \right)^2 + \left(\frac{\partial v'}{\partial x} \right)^2 + \left(\frac{\partial w'}{\partial x} \right)^2 + \left(\frac{\partial u'}{\partial x} \right)^2 \right] &
\end{aligned}$$

This form of the equation is essentially the same as the one given by Rouse(1960). The equation is composed of the following terms:

1. axial and radial convection of turbulent kinetic energy

$$\overline{U} \frac{\partial k}{\partial x} + \overline{V} \frac{\partial k}{\partial r}$$

2. axial diffusion of turbulent kinetic energy

$$\frac{1}{\rho} \frac{\partial}{\partial x} \left[\mu \frac{\partial k}{\partial r} - \rho \frac{\overline{u'v'v'}}{2} - \rho \frac{\overline{u'w'w'}}{2} - \rho \frac{\overline{u'u'u'}}{2} - \overline{u'p'} \right]$$

3. radial diffusion of turbulent kinetic energy

$$\frac{1}{\rho r} \frac{\partial}{\partial r} \left[\mu r \frac{\partial k}{\partial r} - \rho r \frac{\overline{v'v'v'}}{2} - \rho r \frac{\overline{v'w'w'}}{2} - \rho r \frac{\overline{u'u'v'}}{2} - r \overline{v'p'} \right]$$

4. additional terms which are usually very small relative to other terms

$$\left[\frac{2}{\rho} \mu \frac{\overline{v'v'}}{r^2} - \frac{\mu}{\rho} \frac{\overline{w'w'}}{r^2} \right]$$

5. production of turbulent kinetic energy

$$\left[\overline{v'v'} \frac{\partial \overline{V}}{\partial r} + \overline{u'v'} \frac{\partial \overline{U}}{\partial r} - \overline{u'v'} \frac{\partial \overline{V}}{\partial x} + \overline{u'u'} \frac{\partial \overline{V}}{\partial x} + \overline{w'w'} \frac{\overline{V}}{r} \right]$$

6. viscous dissipation of turbulent kinetic energy

$$\frac{\mu}{\rho} \left[\left(\frac{\partial v'}{\partial r} \right)^2 + \left(\frac{\partial w'}{\partial r} \right)^2 + \left(\frac{\partial u'}{\partial r} \right)^2 + \left(\frac{\partial v'}{\partial x} \right)^2 + \left(\frac{\partial w'}{\partial x} \right)^2 + \left(\frac{\partial u'}{\partial x} \right)^2 \right]$$

3.3 Energy Cascade:

Turbulence is an eddying motion which has a wide spectrum of eddy sizes accompanied by wide spectrum of fluctuation frequencies. The largest eddies, which are associated with the low frequency fluctuations, are determined by the boundary conditions of the flow and their sizes are of the same order of magnitude as the flow domain. On the other hand, the smallest eddies, associated with the high frequency fluctuations, are determined by viscous forces. Usually, the large-scale turbulent motions transport momentum and heat and contribute to the turbulence correlations $\overline{u'_i u'_j}$ and $\overline{u'_i \varphi}$ (φ is a scalar such as species concentration). Therefore, the large-scale motion has to be simulated in a turbulence model for the determination of $\overline{u'_i u'_j}$ and $\overline{u'_i \varphi}$. The velocity and length scales introduced in turbulence models are parameters which characterize this motion.

One must note that the energy extracted by large eddies from the mean flow is passed on to smaller and smaller eddies until viscous forces become active and dissipate this energy. This process is called energy cascade.

Because, the mean flow often has preferred directions; therefore, the large-scale turbulent motion can be strongly anisotropic and both the intensity of the fluctuations and their length scales are direction dependent.

3.4 Basic Concepts and Classification of Turbulence Models:

Generally speaking, models which have been developed employ transport equations for quantities characterizing turbulence such as terms representing the convective transport by the mean motion and the diffusion transport by the turbulent motion. The philosophy behind this is that complex turbulence phenomena can be described so that the model becomes more and more universal.

3.4.1 Zero Equation Model: Models here are relatively simple and do not involve transport equations for turbulence modeling. Boussinesq (1877) was the first to propose a turbulence model in which he related the Reynolds stresses $-\rho \overline{u'_i u'_j}$ to mean flow

parameters. He assumed that, in analogy to the viscous stresses in laminar flows, the turbulent stresses are proportional to the mean-velocity gradients. For general flow situations, this may be expressed as:

$$-\rho \overline{u'_i u'_j} = \mu_t \left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \rho k$$

Note that k (turbulent kinetic energy) is defined by

$$k = \frac{1}{2} \sum_{N=1}^3 \overline{u_N'^2}$$

and the Kronecker delta $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$. μ_t is the turbulent or eddy viscosity which is not a fluid property and may vary significantly from one point in the flow to another or from flow to flow. Now, the main problem is shifted to determining the distribution of μ_t .

Prandtl (1925) introduced the mixing length theory. He assumed that ;

$$\mu_t = \rho L^2 \frac{\partial \overline{U}_i}{\partial x_j}$$

L is the mixing length and is related to the geometry of the flowfield. Unfortunately, his assumption is not suitable when the geometry is complex or the mean flow gradient is zero.

3.4.2 One Equation Model: Turbulence models utilizing transport of turbulence quantities were developed and utilized to determine the length scale from them. Kolmogorov (1943) assumed that

$$\mu_t = \rho \sqrt{k} l.$$

This formula is known as Kolmogorov-Prandtl expression because both Kolmogorov and Prandtl introduced it independently. l is a length related to the size of eddies in the flow. To solve the problem, both k and l has to be specified. k could be estimated by solving the turbulent kinetic energy (see Appendix B). The one equation model still has deficiency similar to zero equation model (mixing length theory); therefore, it is restricted to simple flows.

3.4.3 Two Equation Model: A transport equation for l could be developed and utilized to determine l . Two equations of k and l could be solved. Better combinations such as $k - \omega$, $k - kl$ and $k - \epsilon$ (ϵ is the dissipation) are more useful. In fact, $k - \epsilon$ is the most popular of these approaches (see Gould et al., 1990; and Rodi, 1980).

$$\epsilon \propto \frac{k^{\frac{3}{2}}}{l}$$

PARCH code utilized the $k - \epsilon$ model to predict missile base flow (plume flowfield, see Kennedy et al., 1992). PARCH is a fully elliptic, Navier-Stokes solver with finite rate chemistry and a two equation $k - \epsilon$ model. Also, Petrie and Walker 1985 used two equation models to predict missile base flow with a centered propulsive jet.

4 CALCULATION PROCEDURES AND RESULTS

To calculate each term of the turbulence energy equation from experimental data, first and second derivatives have to be evaluated. For example, central difference was used to calculate the first derivatives while the second derivatives were estimated using the method of undetermined coefficients (see Lapidus and Pinder, 1982). This takes care of the non-uniformity in spacing along the x and r axes. The method is also valid for uniform spacing (e.g., when the method of undetermined coefficients is used for equal spacing of points, its result is the same like the conventional method). For example, the discretized expression for the second derivatives of k with respect to a coordinate axis x at a location (i, j) , using the method of undetermined coefficient, is as follows:

$$\frac{\partial^2 k_{i,j}}{\partial x^2} = \alpha_{i+1,j} k_{i+1,j} + \alpha_{i,j} k_{i,j} + \alpha_{i-1,j} k_{i-1,j}$$

$$\text{where } \alpha_{i,j} = \left(\frac{H_1^2}{2} + \frac{H_2 H_1}{2} \right)^{-1},$$

$$\alpha_{i+1,j} = \left(\frac{H_2 H_1}{2} + \frac{H_2^2}{2} \right)^{-1}, \text{ and}$$

$$\alpha_{i-1,j} = -1.0 * (\alpha_{i,j} + \alpha_{i+1,j})$$

$$\text{while, } H_1 = x_i - x_{i-1}, \text{ and } H_2 = x_{i+1} - x_i$$

To apply the balance of the energy, some terms are neglected because they are small and difficult to calculate due to insufficient experimental data. For example, the pressure diffusion terms $\overline{u'p'}$ and $\overline{v'p'}$ were assumed to be negligible, which is a reasonable assumption at locations away from the wall (see Gould et al., 1990). Also, the exact TKE equation contains some terms that were not measured. The following assumptions were made: $\overline{w'^2} = \overline{v'^2}$, $\overline{u'w'^2} = \overline{u'v'^2}$, and $\overline{v'w'^2} = \overline{v'^3}$ (see Logan, 1972 and Azad et al., 1987).

4.1 Mean and Turbulent Flow Data

Measurements of the axial and radial mean velocity and turbulent normal stresses in addition to Reynolds stress, turbulent kinetic energy, and turbulent triple products were made but they will not be presented here since they are classified. The measurements were made at numerous radial locations across the test section and also at numerous axial planes. Upstream velocity profiles were found to be very flat except near the walls of the model. Velocity measurements on both sides of the centerline at several axial locations indicated that the flowfield was symmetric.

4.2 Validation of Modeled Turbulent Stresses

Numerically, the set of governing equations for an incompressible, axisymmetric turbulent flow are a set of five coupled, nonlinear, partial differential equations that are solved numerically with the appropriate boundary conditions for the dependent variables \overline{U} , \overline{V} , \overline{P} , k , and ϵ . Three of the governing equations are the time averaged continuity equation and the axial and radial momentum equations with the viscosity set equal to the effective viscosity. The TEACH code uses the $k - \epsilon$ turbulence model to predict the turbulent momentum transport. The steady state solution of the TEACH code provides the values for the mean axial and radial velocities, \overline{U} and \overline{V} , the pressure \overline{P} , the turbulent kinetic energy k , and the turbulent dissipation ϵ , at each node point in the flow domain. First level validation of the code could be made by directly comparing measured mean velocities with the predicted ones. The

turbulent kinetic energy k can also be compared directly with experimental measurements, giving a second level of validation. An additional level of code validation could be performed to determine how well the $k - \epsilon$ turbulence model estimates the unknown turbulent normal stresses $\overline{u'_j u'_j}$ and the Reynolds stresses $\overline{u'_i u'_j}$. This validation is performed by reversing the modeling steps used in the $k - \epsilon$ turbulence closure procedure. The closure procedure is as follows:

The turbulent viscosity μ_t , in the $k - \epsilon$ model is given by

$$\mu_t = C_D \rho k^2 / \epsilon$$

C_D is an empirical coefficient. Thus, in addition to the preceding five dependent variables, the value of μ_t , is also known throughout the computational flow domain. The unknown turbulent normal and shear stresses are then found using μ_t , and the mean velocity gradients through the modified Boussinesq approximation for constant density fluid (see Gould et al., 1990 for more details). These modeled turbulent stresses can now be compared directly with the measured values giving a higher level of code validation. Similar studies (but for different applications) were made using these concepts; such as Wagniski and Fiddler, 1965; Driver and Seegmiller, 1985; Moore et al., 1994^a, and 1994^b; and Pinarbasi and Johnson, 1995.

5 SUMMARY AND CONCLUSIONS

A two component LDV was used to measure accurately \overline{U} , \overline{V} , $\overline{u'^2}$, $\overline{u'v'}$, $\overline{u'^3}$, $\overline{u'^2 v'}$, $\overline{u' v'^2}$, and $\overline{v'^3}$ in the flowfield downstream of the base of a generic tactical missile configuration in simulated powered flight. The data are unique in their details, high quality and accuracy, no other similar data are available. The measured values of \overline{U} , \overline{V} , $\overline{u'^2}$, $\overline{v'^2}$, $\overline{u'^2}$, and turbulent kinetic energy k are examined but not reported here due to its classification. Turbulent triple products were also examined and found to contribute significantly to the transport of turbulent kinetic energy. A balance of the turbulent kinetic energy identified the various areas of the flow where different turbulent transport mechanisms dominate. Examinations of the non-isotropy factor expose one of the major failings of the modified Boussinesq approximation, namely that it does not properly account for anisotropy of the flow.

6 ACKNOWLEDGMENTS

The author would like to acknowledge and thank Mr. F. Helstley for use of his data and for all the help he provided. Thanks are also due to Dr. W. Ruyten and Mr. W. D. Williams for their support in the laboratory. This investigation was performed at Arnold Engineering and Development Center (MC/AEDC) under the Summer Faculty Research Program supported by AFOSR.

References

- [1] Ahmed, S. A., Rose, A., and Nejad, A. S., (1992) "Three Component LDV Velocity Measurements in a Can Type Research Combustor for CFD Validation Part 1. Isothermal Flow," ASME 92-GT-138.
- [2] Azad, R. S., Kassab, S. Z., and Dang, T. H., (1987) "Experimental Evaluation of Approximations for $\overline{\omega'^2 v'}$ and $\overline{u' \omega'^2}$ " AIAA Journal, Vol. 25, pp. 171-173.
- [3] Driver, D. M. and Seegmiller, H L., (1985) "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow," AIAA Journal, Vol . 23, pp. 163-171.
- [4] Dutton, J.C., Herrin, J.L., Molezzi, M.J., Mathur, T. and Smith, K.M. (1995) "Recent Progress on High-Speed Separated Base Flows" AIAA paper 95-0472.
- [5] Gould, R.D, Stevenson, W.H. and Thompson, H.D. (1990) "Investigation of Turbulent Transport in an Axisymmetric Sudden Expansion," AIAA Journal, vol. 28, pp. 276-283.
- [6] Harlow, F. H. and Nakayama, P. 1., (1968) "Transport of Turbulence Energy Decay Rate," Los Alamos Science Laboratory, University of California, Rept. LA 3854.
- [7] Hinze, J.O., (1975), "Turbulence", 2nd ed., Mc GrawHill, New York.
- [8] Kennedy, K. D., Walker, B. J., Mikkelsen, C. D., and Heltsley, F. L., (1992) "Measurements and Predictions for Velocity and Turbulent Kinetic Energy in Air/Air Plumes." AIAA paper, AIAA 92-2782.

- [9] Kolmogorov, A.N., (1942) "Equations of Turbulent Motion of an Incompressible Fluid," *Izv. Akad. Nauk. SSR, Seria Fizicheskaya*, No. 1-2, pp. 56-58.
- [10] Lapidus, L., and Pinder, G.F., (1982) "Numerical Solution of Partial Differential Equations in Science and Engineering", John Wiley and Sons, New York.
- [11] Launder, B. E., Reece, G. J., and Rodi, W., (1975) "Progress in the Development of a Reynolds Stress Turbulence Closure," *Journal of Fluid Mechanics*, Vol. 68, pp. 537-566.
- [12] Logan, S. E., (1972) "A Laser Velocimeter for Reynolds Stresses and Other Turbulence Parameters," *AIAA Journal*, Vol. 19, pp. 933-935.
- [13] Moore, J., Moore, J., Heckel, S., and Ballesteros, R., (1994^a) "Reynolds Stresses and Dissipation Mechanisms in a Turbine Tip Leakage Vortex," ASME paper 94-GT-267.
- [14] Moore, J., (1994^b) "Osborne Reynolds: Energy Methods in Transition and Loss production - A Centennial Perspective" ASME 94-GT-225.
- [15] Petrie, H.L. and Walker B.J. (1985), "Comparison of Experiment and Computation for a missile Base Flowfield with a Central Propulsive Jet," AIAA paper 85-1618.
- [16] Pinarbasi, A, and Johnson M., (1995) "A Study of Reynolds Stresses and Loss Generation at the inlet to a Centrifugal Vaneless Diffuser" ASME 95-GT-150.
- [17] Prandtl, L., (1945) "Über ein neues Formelsystem für die ausgebildete Turbulenz," *Nachr. Akad. Wiss., Göttingen, Math.-Phys. Klasse*, pp. 6.
- [18] Prandtl, L., (1925) "Über die ausgebildete Turbulenz," *ZAMM*, Vol. 5, pp. 36.
- [19] Reynolds, O., (1895), "On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion," *Phil. Trans. Roy. Soc. A.*, Vol. 186, pp. 123-164.
- [20] Rodi, W., (1980) "Turbulence Models and Their Application in Hydraulics - A state of the Art Review," International Association for Hydraulic Research.
- [21] Rouse, H., Yih, C. S. and Humphreys, H.W., (1952) "Gravitational Convection from a boundary Source," *Tellus*, 4, pp. 201-210.

- [22] Tennekes, H. and Lumeley, J.,(1972)" A First Course in Turbulence", The MIT Press, Cambridge, Mass.
- [23] Walker, B. J., and Heltsley, F. L., (1993) "Nonintrusive Diagnostics of Supersonic Missile Afterbody Flows." 15th ICIASF 93, st. Louis, France
- [24] Wygnanski, I and Fiedler,(1969)"Some Measurements in the Self-Preserving Jet," Journal of Fluid Mechanics, vol.38, pp. 577.

APPENDIX A

Reynolds Equation

Assume U , V , and W are the velocity components in the directions of the cylindrical coordinates x , r , and θ ; respectively. The conservation of mass is

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho U)}{\partial x} + \frac{1}{r} \frac{\partial (r \rho V)}{\partial r} + \frac{1}{r} \frac{\partial (\rho W)}{\partial \theta} = 0 \quad (1)$$

The conservation of momentum equations in the three coordinate directions are:

1. Axial momentum equation:

$$\begin{aligned} \frac{\partial (\rho U)}{\partial t} + \frac{\partial (\rho U^2)}{\partial x} + \frac{1}{r} \frac{\partial (r \rho UV)}{\partial r} + \frac{1}{r} \frac{\partial (\rho UW)}{\partial \theta} = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x} \left[2\mu \frac{\partial U}{\partial x} - \frac{2}{3}\mu \phi \right] + \\ \frac{1}{r} \frac{\partial}{\partial r} \left[r\mu \left(\frac{\partial U}{\partial r} + \frac{\partial V}{\partial x} \right) \right] + \frac{1}{r} \frac{\partial}{\partial \theta} \left[\mu \left(\frac{\partial V}{\partial x} + \frac{1}{r} \frac{\partial U}{\partial \theta} \right) \right] \end{aligned} \quad (2)$$

2. Radial momentum equation:

$$\begin{aligned} \frac{\partial (\rho V)}{\partial t} + \frac{\partial (\rho VU)}{\partial x} + \frac{1}{r} \frac{\partial (r \rho V^2)}{\partial r} + \frac{1}{r} \frac{\partial (\rho VW)}{\partial \theta} - \frac{\rho V^2}{r} = -\frac{\partial P}{\partial r} + \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial U}{\partial r} + \frac{\partial V}{\partial x} \right) \right] + \\ \frac{1}{r} \frac{\partial}{\partial r} \left[r\mu \left(2\frac{\partial V}{\partial r} - \frac{2}{3}\phi \right) \right] + \frac{1}{r} \frac{\partial}{\partial \theta} \left[\mu \left(r \frac{\partial V}{\partial x} + \frac{1}{r} \frac{\partial V}{\partial \theta} \right) \right] - \frac{\mu}{r} \left[2\frac{\partial U}{\partial x} + \frac{2V}{r} - \frac{2}{3}\phi \right] \end{aligned} \quad (3)$$

3. Tangential momentum equation:

$$\begin{aligned} \frac{\partial (\rho W)}{\partial t} + \frac{\partial (\rho WU)}{\partial x} + \frac{1}{r} \frac{\partial (r \rho WV)}{\partial r} + \frac{1}{r} \frac{\partial (\rho W^2)}{\partial \theta} - \frac{\rho WV}{r} = -\frac{1}{r} \frac{\partial P}{\partial \theta} + \\ \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial W}{\partial x} + \frac{1}{r} \frac{\partial U}{\partial \theta} \right) \right] + \frac{1}{r^2} \frac{\partial}{\partial r} \left[r^2 \mu \left(r \frac{\partial W}{\partial r} - \frac{1}{r} \frac{\partial V}{\partial \theta} \right) \right] + \frac{1}{r} \frac{\partial}{\partial \theta} \left[2\mu \left(\frac{1}{r} \frac{\partial W}{\partial \theta} + \frac{V}{r} \right) - \frac{2}{3}\phi \right] \end{aligned} \quad (4)$$

ϕ is defined to be

$$\frac{\partial U}{\partial x} + \frac{1}{r} \frac{\partial}{\partial r} (rV) + \frac{1}{r} \frac{\partial W}{\partial \theta}$$

Reynolds defined the instantaneous values of velocity, pressure, and density to be composed of a mean value and a fluctuating component as follows:

$$U = \bar{U} + u',$$

$$V = \bar{V} + v',$$

$$W = \bar{W} + w',$$

$$P = \bar{P} + p', \text{ and}$$

$$\rho = \bar{\rho} + \rho' \quad (5)$$

Substitute with these values in the conservation of mass to get the time averaged continuity equation (time average is performed):

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x} (\bar{\rho} \bar{U} + \overline{\rho' u'}) + \frac{1}{r} \frac{\partial}{\partial r} (r \bar{\rho} \bar{V} + r \overline{\rho' v'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\bar{\rho} \bar{W} + \overline{\rho' w'}) = 0 \quad (6)$$

Similarly, the time averaged Reynolds equations for turbulent flow in cylindrical coordinates may be obtained by substituting the above values into momentum equations and time averaging to get the following:-

1. Time averaged axial Reynolds equation:

$$\begin{aligned} & \frac{\partial}{\partial t} (\bar{\rho} \bar{U}) + \frac{1}{r} \frac{\partial}{\partial r} (r \bar{\rho} \bar{V} \bar{U}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\bar{\rho} \bar{W} \bar{U}) + \frac{\partial}{\partial x} (\bar{\rho} \bar{U} \bar{U}) + \\ & \left[\frac{1}{r} \frac{\partial}{\partial r} (r \overline{\rho' v' u'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{\rho' w' u'}) + \frac{\partial}{\partial x} (\overline{\rho' u' u'}) \right] + \\ & \left[\frac{1}{r} \frac{\partial}{\partial r} (r \bar{U} \overline{\rho' v'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\bar{U} \overline{\rho' w'}) + \frac{\partial}{\partial x} (\bar{U} \overline{\rho' u'}) \right] + \\ & \left[\frac{1}{r} \frac{\partial}{\partial r} (r \bar{V} \overline{\rho' u'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\bar{W} \overline{\rho' u'}) + \frac{\partial}{\partial x} (\bar{U} \overline{\rho' u'}) \right] + \\ & \frac{\partial}{\partial t} (\overline{\rho' u'}) + \frac{1}{r} \frac{\partial}{\partial r} (r \overline{\rho' v' u'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{\rho' w' u'}) + \frac{\partial}{\partial x} (\overline{\rho' u' u'}) = -\frac{\partial \bar{P}}{\partial x} + \\ & \frac{1}{r} \frac{\partial}{\partial r} \left[r \mu \left(\frac{\partial \bar{U}}{\partial r} + \frac{\partial \bar{V}}{\partial x} \right) \right] + \frac{1}{r} \frac{\partial}{\partial \theta} \left[\mu \left(\frac{\partial \bar{W}}{\partial x} + \frac{1}{r} \frac{\partial \bar{U}}{\partial \theta} \right) \right] + \frac{\partial}{\partial x} \left[2\mu \frac{\partial \bar{U}}{\partial x} - \frac{2}{3} \mu \phi \right] \quad (7) \end{aligned}$$

2. Time averaged radial Reynolds equation:

$$\begin{aligned} & \frac{\partial}{\partial t} (\bar{\rho} \bar{V}) + \frac{1}{r} \frac{\partial}{\partial r} (r \bar{\rho} \bar{V} \bar{V}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\bar{\rho} \bar{W} \bar{V}) + \frac{\partial}{\partial x} (\bar{\rho} \bar{U} \bar{V}) - \frac{\bar{\rho} \bar{W} \bar{W}}{r} + \\ & \left[\frac{1}{r} \frac{\partial}{\partial r} (r \overline{\rho' v' v'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{\rho' w' v'}) + \frac{\partial}{\partial x} (\overline{\rho' u' v'}) - \frac{\overline{\rho' w' w'}}{r} \right] + \\ & \left[\frac{1}{r} \frac{\partial}{\partial r} (r \bar{V} \overline{\rho' v'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\bar{V} \overline{\rho' w'}) + \frac{\partial}{\partial x} (\bar{V} \overline{\rho' u'}) - \frac{\overline{\rho' w' w'}}{r} \right] + \\ & \left[\frac{1}{r} \frac{\partial}{\partial r} (r \bar{V} \overline{\rho' v'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\bar{W} \overline{\rho' v'}) + \frac{\partial}{\partial x} (\bar{U} \overline{\rho' v'}) - \frac{\overline{\rho' w' w'}}{r} \right] + \end{aligned}$$

$$\begin{aligned}
& \frac{\partial}{\partial t} (\overline{\rho'v'}) + \frac{1}{r} \frac{\partial}{\partial r} (r \overline{\rho'v'v'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{\rho'w'v'}) + \frac{\partial}{\partial x} (\overline{\rho'u'v'}) - \frac{\overline{\rho'w'w'}}{r} = -\frac{\partial \overline{P}}{\partial r} + \\
& \frac{1}{r} \frac{\partial}{\partial r} \left[r \mu \left(2 \frac{\partial \overline{V}}{\partial r} - \frac{2}{3} \phi \right) \right] + \frac{1}{r} \frac{\partial}{\partial \theta} \left[\mu \left(r \frac{\partial \overline{V}}{\partial x} + \frac{1}{r} \frac{\partial \overline{V}}{\partial \theta} \right) \right] - \frac{\mu}{r} \left[2 \frac{\partial \overline{U}}{\partial x} + \frac{2 \overline{V}}{r} - \frac{2}{3} \phi \right] + \\
& \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial \overline{U}}{\partial r} + \frac{\partial \overline{V}}{\partial x} \right) \right] \quad (8)
\end{aligned}$$

3. Time averaged tangential Reynolds equation:

$$\begin{aligned}
& \frac{\partial}{\partial t} (\overline{\rho W}) + \frac{1}{r} \frac{\partial}{\partial r} (r \overline{\rho V W}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{\rho W W}) + \frac{\partial}{\partial x} (\overline{\rho U W}) + \frac{\overline{\rho V W}}{r} + \\
& \left[\frac{1}{r} \frac{\partial}{\partial r} (r \overline{\rho v'w'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{\rho w'w'}) + \frac{\partial}{\partial x} (\overline{\rho u'w'}) + \frac{\overline{\rho v'w'}}{r} \right] + \\
& \left[\frac{1}{r} \frac{\partial}{\partial r} (r \overline{W \rho'v'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{W \rho'w'}) + \frac{\partial}{\partial x} (\overline{W \rho'u'}) + \frac{\overline{\rho'v'W}}{r} \right] + \\
& \left[\frac{1}{r} \frac{\partial}{\partial r} (r \overline{V \rho'w'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{W \rho'w'}) + \frac{\partial}{\partial x} (\overline{U \rho'w'}) + \frac{\overline{\rho'w'V}}{r} \right] + \\
& \frac{\partial}{\partial t} (\overline{\rho'w'}) + \frac{1}{r} \frac{\partial}{\partial r} (r \overline{\rho'w'w'}) + \frac{1}{r} \frac{\partial}{\partial \theta} (\overline{\rho'w'w'}) + \frac{\partial}{\partial x} (\overline{\rho'u'w'}) - \\
& \frac{\overline{\rho'v'w'}}{r} = -\frac{1}{r} \frac{\partial \overline{P}}{\partial \theta} + \frac{1}{r^2} \frac{\partial}{\partial r} \left[r^2 \mu \left(r \frac{\partial \overline{W}}{\partial r} - \frac{1}{r} \frac{\partial \overline{V}}{\partial \theta} \right) \right] + \frac{1}{r} \frac{\partial}{\partial \theta} \left[2 \mu \left(\frac{1}{r} \frac{\partial \overline{W}}{\partial \theta} + \frac{\overline{V}}{r} \right) - \frac{2}{3} \phi \right] + \\
& \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial \overline{W}}{\partial x} + \frac{1}{r} \frac{\partial \overline{U}}{\partial \theta} \right) \right] \quad (9)
\end{aligned}$$

The Reynolds equations in tensor form could be written as:

$$\begin{aligned}
& \frac{\partial}{\partial t} (\overline{\rho U_i}) + \frac{\partial}{\partial t} (\overline{\rho' u'_i}) + \frac{\partial}{\partial x_j} (\overline{\rho U_i U_j}) + \frac{\partial}{\partial x_j} (\overline{U_i \rho' u'_j}) + \frac{\partial}{\partial x_j} (\overline{U_j \rho' u'_i}) + \frac{\partial}{\partial x_j} (\overline{\rho u'_i u'_j}) + \\
& \frac{\partial}{\partial x_j} (\overline{\rho' u'_i u'_j}) = -\frac{\partial \overline{P}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial \overline{U_i}}{\partial x_j} + \frac{\partial \overline{U_j}}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial \overline{U_k}}{\partial x_k} \right) \right] \quad (10)
\end{aligned}$$

For constant density flow (or neglecting the fluctuations in the density), the above equation reduces to the following form:-

$$\frac{\partial}{\partial t} (\overline{\rho U_i}) + \frac{\partial}{\partial x_j} (\overline{\rho U_i U_j}) + \frac{\partial}{\partial x_j} (\overline{\rho u'_i u'_j}) = -\frac{\partial \overline{P}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \frac{\partial \overline{U_i}}{\partial x_j} \right] + \frac{\partial \overline{U_j}}{\partial x_i} \frac{\partial \mu}{\partial x_j} \quad (11)$$

if cartesian coordinates were used instead of cylindrical coordinates, these new coordinates could be utilized:

$$x = x, \quad y = r \quad \text{and} \quad z = r\theta$$

and the equations in rectangular coordinates could be evaluated.

APPENDIX B

Turbulence Energy Equation

The turbulence energy equation could be derived by multiplying the *ith* Reynolds equation by u'_i ($i = 1, 2$ and 3) and then summing all the equations. The new equation in tensor form is as follows:-

$$\begin{aligned} & \overline{\rho' u'_i} \frac{\partial \bar{U}_i}{\partial t} + \frac{\partial}{\partial t} (\bar{\rho} k) + \frac{\partial}{\partial t} (\bar{\rho}' k) + \frac{\partial}{\partial x_j} (\bar{\rho} \bar{U}_j k) + \frac{\partial}{\partial x_j} (\bar{U}_j \bar{\rho}' k) + \\ & \left[\frac{\partial}{\partial x_j} (\bar{\rho} u'_j k) + \frac{\partial}{\partial x_j} (\bar{\rho}' u'_j k) + \bar{\rho} u'_i u'_j \frac{\partial \bar{U}_i}{\partial x_j} + \bar{\rho}' u'_i \bar{U}_j \frac{\partial \bar{U}_i}{\partial x_j} \right] + \bar{\rho}' u'_i u'_j \frac{\partial \bar{U}_i}{\partial x_j} = - \frac{\partial}{\partial x_i} (\bar{u}'_i \bar{\rho}') + \bar{\rho}' \frac{\partial u'_i}{\partial x_i} + \\ & \frac{\partial}{\partial x_j} \left[\mu \frac{\partial}{\partial x_j} (k) \right] - \mu \left[\frac{\partial u'_i}{\partial x_j} \right]^2 + \overline{u'_i \frac{\partial u'_j}{\partial x_i} \frac{\partial \mu}{\partial x_j}} + \frac{1}{3} \overline{\mu u'_i \frac{\partial}{\partial x_i} \left[\frac{\partial u'_k}{\partial x_k} \right]} - \frac{2}{3} \overline{u'_j \frac{\partial \mu}{\partial x_j} \left[\frac{\partial u'_k}{\partial x_k} \right]} \end{aligned} \quad (1)$$

For constant density flow (or neglecting the fluctuations in the density), the above equation becomes:

$$\begin{aligned} & \frac{\partial}{\partial t} (\bar{\rho} k) + \frac{\partial}{\partial x_j} (\bar{\rho} \bar{U}_j k) + \frac{\partial}{\partial x_j} (\bar{\rho} u'_j k) + \bar{\rho} u'_i u'_j \frac{\partial \bar{U}_i}{\partial x_j} = - \frac{\partial}{\partial x_i} (\bar{u}'_i \bar{\rho}') + \frac{\partial}{\partial x_j} \left[\mu \frac{\partial}{\partial x_j} (k) \right] - \\ & \mu \left[\frac{\partial u'_i}{\partial x_j} \right]^2 + \overline{u'_i \frac{\partial u'_j}{\partial x_i} \frac{\partial \mu}{\partial x_j}} \end{aligned} \quad (2)$$

This equation in cylindrical coordinates (for steady, axisymmetric, and constant density flow) could be written as

$$\begin{aligned} & \bar{U} \frac{\partial k}{\partial x} + \bar{V} \frac{\partial k}{\partial r} = \frac{1}{\rho} \frac{\partial}{\partial x} \left[\mu \frac{\partial k}{\partial x} - \rho \frac{\overline{u'v'v'}}{2} - \rho \frac{\overline{u'w'w'}}{2} - \rho \frac{\overline{u'u'u'}}{2} - \bar{u}' \bar{p}' \right] + \\ & \frac{1}{\rho r} \frac{\partial}{\partial r} \left[\mu r \frac{\partial k}{\partial r} - \rho r \frac{\overline{v'v'v'}}{2} - \rho r \frac{\overline{v'w'w'}}{2} - \rho r \frac{\overline{u'u'v'}}{2} - r \bar{v}' \bar{p}' \right] - \left[\frac{2}{\rho} \mu \frac{\overline{v'v'}}{r^2} - \frac{\mu}{\rho} \frac{\overline{w'w'}}{r^2} \right] - \\ & \left[\overline{v'v'} \frac{\partial \bar{V}}{\partial r} + \overline{u'v'} \frac{\partial \bar{U}}{\partial r} + \overline{u'v'} \frac{\partial \bar{V}}{\partial x} + \overline{u'u'} \frac{\partial \bar{V}}{\partial x} + \overline{w'w'} \frac{\bar{V}}{r} \right] - \\ & \frac{\mu}{\rho} \left[\left(\frac{\partial v'}{\partial r} \right)^2 + \left(\frac{\partial w'}{\partial r} \right)^2 + \left(\frac{\partial u'}{\partial r} \right)^2 + \left(\frac{\partial v'}{\partial x} \right)^2 + \left(\frac{\partial w'}{\partial x} \right)^2 + \left(\frac{\partial u'}{\partial x} \right)^2 \right] \end{aligned} \quad (3)$$

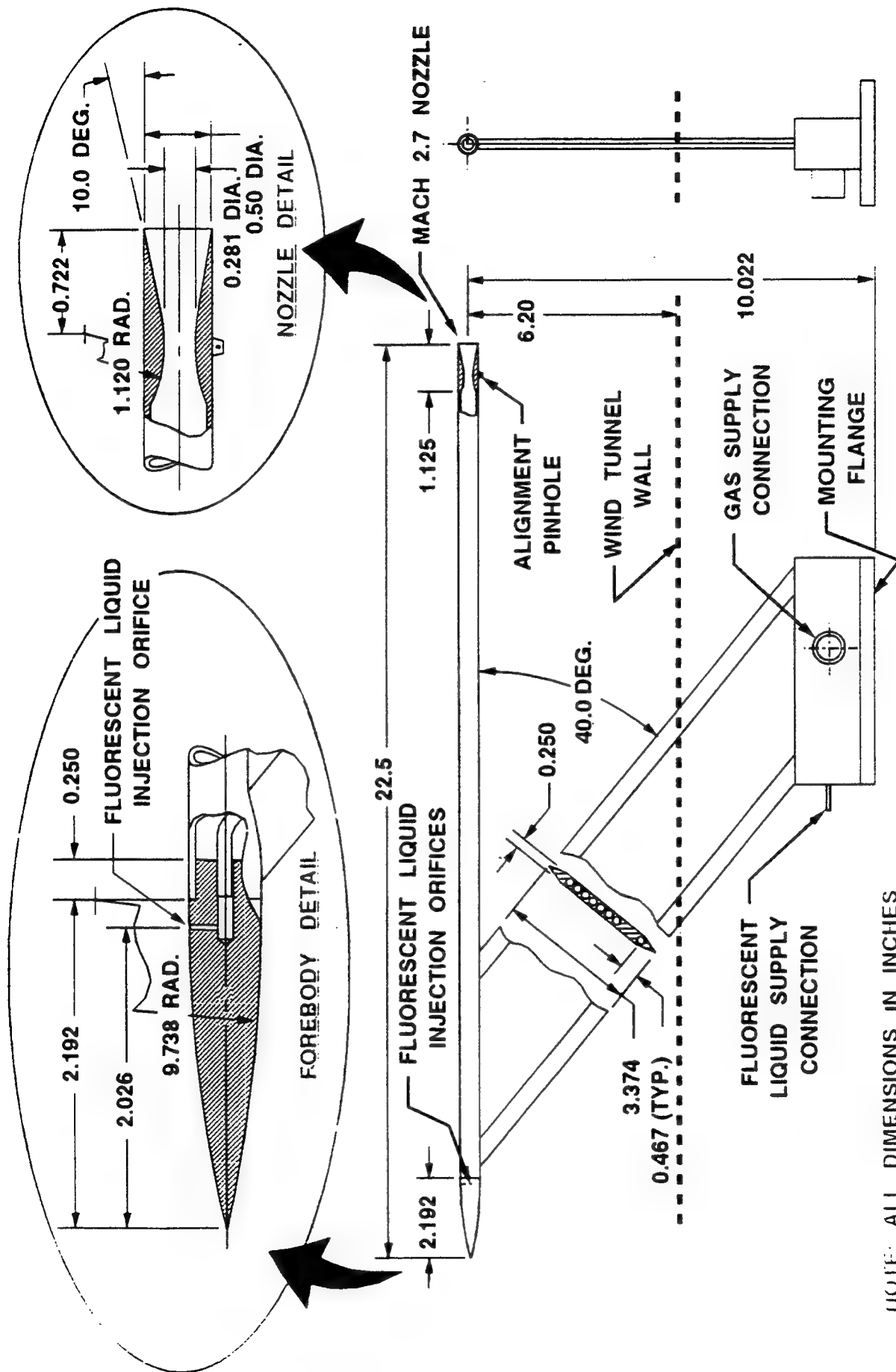


Fig. 1 Schematic of Small Generic Zero-Base Missile Model

MODEL-BASED SYSTEM DEVELOPMENT
FOR
AEROSPACE TESTING INSTRUMENTATION

Dr. Theodore A. Bapty
Research Assistant Professor

James R. Davis
Graduate Student
Department of Electrical and Computer Engineering

Vanderbilt University
400 24th Avenue South
Nashville, TN 37235

Summer Faculty/Graduate Student Research Program
held at
Arnold Engineering Development Center

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.

September 1993

MODEL-BASED SYSTEM DEVELOPMENT
FOR
AEROSPACE TESTING INSTRUMENTATION

Theodore A. Bapty
Research Assistant Professor

James R. Davis
Graduate Student
Department of Electrical and Computer Engineering

Abstract

The complexity of modern aerospace systems requires significant amounts of testing to ensure correct operation and cost effective development. Ground testing enables extremely accurate control over the external conditions along with the ability to acquire a large number of measurements of internal engine structures.

Analysis and verification of turbine engine testing data is a computationally intensive, dynamic process. The combination of high bandwidths, large numbers of channels, and constantly changing processing requirements place many demands on an instrumentation and data analysis system. The complexity of building and managing the data systems necessary for the analysis/verification requirement is a difficult task, considering the real-time, parallel, and reactive characteristics of these systems.

Several systems have been developed at AEDC using **Model-Based** system synthesis techniques. This report describes three systems in various stages of development. The Computer Assisted Dynamic Data Acquisition and Monitoring System(CADDMAS), developed in conjunction with Arnold Engineering Development Center, has been in operation for two years in support of simulated altitude turbine engine stress testing. The system uses parallel processing to sustain over 800 million floating point operations per second(MFLOPS) producing on-line data analysis plots, delivering true supercomputer performance. Various improvements to the system have been implemented during the program.

The Data Validation (DATVAL) system, currently entering operation, monitors turbine engine performance data to verify data integrity. Errors are detected in the processed results and the sources of these errors are traced back to the input data, thus diagnosing the original problem. Methods for automatically generating the diagnosis information have been developed during the summer program.

A monitoring and diagnostics system for the large electric motors that power the testing facilities is currently in the initial design phase. The preliminary specifications for the system are currently under development.

The diverse systems described are developed using a common model-based program synthesis technique. These techniques benefit from software reuse and rapid application development. These factors greatly reduce the cost of the development efforts and decrease the design cycle time.

MODEL-BASED SYSTEM DEVELOPMENT FOR AEROSPACE TESTING INSTRUMENTATION

Theodore A. Bapty
James R. Davis

Introduction

During the development and operational lifespan of a turbine engine design numerous tests are performed to ensure design integrity and performance over time. Aerospace ground testing forms a basis for this evaluation process. The engine is installed in a controlled environment test cell that simulates a wide variety of conditions. The conditions are adjusted to approximate the entire operating envelope of the engine, determining the operation of the engine in the field.

Turbine Engine testing generates massive volumes of data at very high rates. A single engine may have several hundred sensors, with bandwidths up to tens of kilohertz. Test periods last for several hours. The entire bulk of data generated is on the order of terabytes for a single test.

Complete analysis of all acquired data has been technically impossible in the past, due to the speed and capacity of standard computers. Historically, only a small fraction of the data has been fully processed by periodic sampling of the data stream. This analysis was also performed off-line, with delays of days to weeks before results were available. This latency prohibits the use of results to guide the testing process.

On-line processing of these signals is critical to improving the testing process. The results from the on-line computations should be available to enable mid-test decisions and interactive test planning. Furthermore, on-line examination of information can assist in catastrophe avoidance, by enabling the detection of pathological engine conditions.

Performing these computations and displaying the results in an interactive format requires an extremely high processing rate. Typical algorithms include frequency transformation, peak detection, validation and correlation. For the analysis of 48 channels of stress data, approximately

800 MFLOPS of sustained computation is required. To process the hundreds of channels available in a typical test, an aggregate sustained computation rate in the GFLOPS range is necessary to meet requirements. The situation is further exacerbated by the requirement to acquire the volumes of data into the computer and to plot the results quickly. The aggregate data input is in the tens of megawords per second. Compounding these requirements, is the real-time requirement that the system must keep up with the data stream and display results interactively.

The majority of algorithms required for turbine engine testing are applied independently on each of the channels. The high computational requirements and natural parallelism make parallel processing a cost-effective hardware approach. The data streams are physically separate channels, so a parallel I/O system is also applicable.

To achieve the multi-GFLOPS goal, a unique computer architecture is required. The Computer Assisted Dynamic Data Monitoring and Analysis System (CADDMAS) has been developed to address these goals. The CADDMAS is a real-time, massively parallel processing system designed to process turbine engine stress data on-line. Results are available immediately, with system outputs defined interactively.

The massive volumes of data are recorded for post processing to assess engine performance and plan future system modifications. The integrity of the data from a test is of paramount importance to the success of the engine development. Hardware problems and configuration problems can result in data errors, corrupting the recorded data, eliminating the benefit of testing. Correcting these problems after the test is expensive, sometimes resulting in repetition of testing programs, at great fiscal and scheduling cost.

Consequently, the on-line verification of engine data is of utmost importance. The Data Validation (DATVAL) system, currently entering operation, monitors turbine engine performance data to verify data integrity. Errors are detected in the processed results and the sources of these errors are traced back to the input data, thus diagnosing the original problem. Representing the information needed for an accurate diagnosis is an extremely difficult task. Methods for automatically generating the diagnosis information have been developed during the summer program.

The facilities required to duplicate the necessary conditions to simulate high altitude flight are immense. Large compressors are required to pump large volumes air into and out of the turbine engine test cell. These compressors are driven by large, 30,000-80,000 horsepower electric motors. The motors have a complicated control system for startup and operation. These motors represent a significant capital expense and are critical to the mission of AEDC. Malfunctions in these motors

result in loss of testing revenue. Damage to the motors results in a large capital outlay and a long system downtime.

When the motors fail to start, a difficult diagnosis procedure ensues. This procedure relies heavily on the experience of small group of irreplaceable engineers and often takes several hours.

Predictive diagnosis could save significant resources, if a impending component failure can be anticipated. Preventive maintenance can concentrate on upcoming problems, reducing the occurrence of unexpected system failures. Preventive maintenance can also reduce the frequency of failure of expensive systems.

A diagnosis and predictive analysis system is currently being defined for use at AEDC. The approaches and capabilities are being considered.

While the hardware needed to meet these testing requirements is available for these systems, and the technology allows easy configuration of arbitrary topologies, the software technology is not as simple. The software is complicated by several factors:

1. **Complex Requirements:** The systems requirements consist of a complex set of processing specifications. The sheer number of specifications and their interrelations results in a significant management task.
2. **Changing Requirements:** The systems requirements change often. The changes must be integrated into the system rapidly.
3. **Distributed Control:** Instead of having one thread of control executing in the system, as many threads as there are processors are used in the system. The problem lies in getting all of these threads to cooperate and accomplish a single task.
4. **Communication:** As the computational data dependencies are divided across processors, there must be a method for transmitting intermediate results from one processor to another.
5. **Real-Time:** The data is coming into the system at a constant rate. The system must respond to all of the data at this rate, or data will be lost.
6. **Dynamically Changing Requirements:** The processing requested from the system is constantly changing. Each test point has a different mix of required output processing.
7. **Modularity:** Systems of various sizes and processing capacity are required. The components (hardware and software) must be designed to allow modules to be assembled quickly and easily

to form arbitrary systems.

8. **Cost:** Systems must be constructed for little more than the hardware cost, due to the cost of the high performance hardware needed to meet system performance goals. Historically, software accounts for 3/4 of the cost of a system like the CADDMAS.

These problems must be solved in a way that will not make the system implementation cost-prohibitive to implement. The next section describes the Model-Based synthesis approach. This discussion is followed by a description of the CADDMAS, the DATVAL, and the prototype engine diagnosis.

Model-Based Synthesis: Modeling the Problem

Multiple-aspect model-based systems offer significant advantages in the management of large-scale, complex systems:

1. Modeling allows abstractions of details into understandable concepts. The model can show only the important information for the current mode of operation. Low-level details can be hidden from the user, and system information can be extracted from these models using information about the modeling domain.
2. Multiple aspect model representation allows the interaction with the model from different perspectives at different times. The user only has to deal with the current mode.
3. Hierarchy in the models enables the hiding of structural details. Only relevant information is visible at any one level in the hierarchy. Complex structures can be represented by a single icon. The user can traverse this hierarchy to examine the system in greater detail.

Domain-specific modeling extends these benefits. The capability to adapt the model representation schema to the application domain further simplifies the problem by presenting familiar concepts to the user. Large systems can still be modeled, but the complexity is hidden in abstractions, alternate aspects, and/or hierarchies.

The selection of the modeling concepts and representation schemas is a key factor in the success of a model-based system. The development of efficient concepts requires a detailed knowledge of

the application domain, the model representation methods, and the interpretation process that maps between the models and the executable system.

The modeling specification and design process begins with the recognition of responsibilities and capabilities of the future system users. Typically, several classes of users will interact with the system. For example, consider the case of the parallel instrumentation domain. The following classes of users are apparent:

- **Instrumentation System Engineers(ISE)** are responsible for development of inherent system capabilities. These include low-level software development, algorithm implementation, and configuration of all system inputs and outputs. The ISE has in-depth knowledge of signal processing, software development, and hardware configuration. These constraints can be relaxed. For instance, when the software is taken from a reusable library, software development expertise is not required.
- The **Test Engineer** is responsible for configuring a working system and handling rapidly changing specifications. The specifications can be changed up to the point of system use, and possibly even during system operation. The test engineer has a deep understanding of the application, and can determine the proper set of requirements to place on the instrumentation system. The test engineer, however, has little knowledge of digital signal processing, software engineering, parallel processing, and embedded systems. This user deals with the concepts of number of system inputs, number and capabilities of system outputs, and requirements for system performance (timing, precision, etc).
- The **Test Operator** actually uses the system, gathering outputs and interpreting the results. This user sees the instrumentation system as a pre-configured black box that generates useful information. The Operator deals with the concepts of a Graphical User Interface for easy access to data, previously configured analysis scenarios, and, most importantly, analysis of the Instrumentation system's output.

The structure of a successful modeling paradigm must recognize the environment: inputs, outputs, system architecture, and the user classes. In this pursuit, the architecture for model-based synthesis of parallel instrumentation systems has been developed. The structure of this paradigm is seen in figure 1.

The Instrumentation Systems Engineers interact with the system to define the library of possible system outputs, using the System Capability Modeling Environment. For each type of analysis,

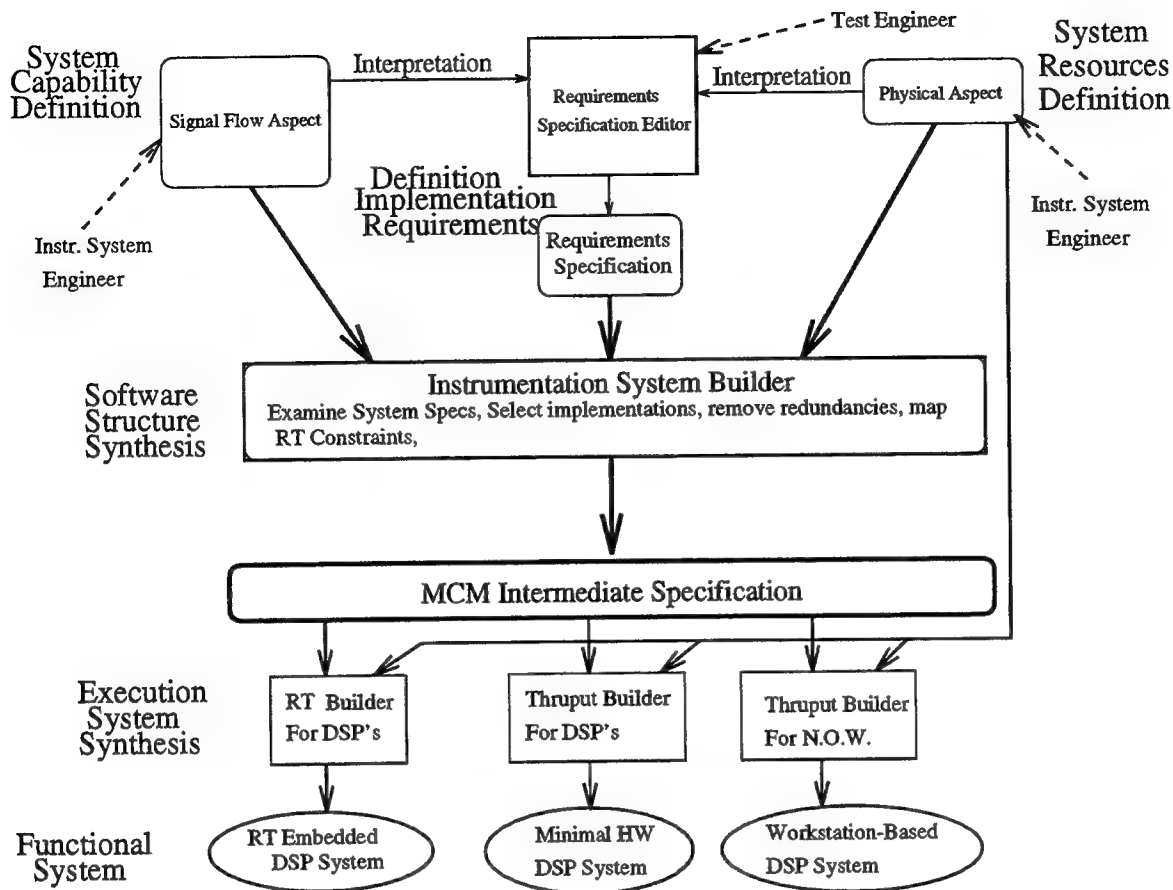


Figure 1: Real-Time Instrumentation System Synthesis Environment

models are defined representing the output type, the various algorithmic implementations of the analysis (in terms of a set of signal flow graphs), and a method for selecting and configuring the various implementations. These signal flow graphs are linked to the source code implementations of the algorithms.

The System Resource Modeling Environment allows modeling of the available input signals, hardware resources, and user interfaces available for system interaction.

These models can be interpreted to create a third modeling environment. This implementation requirements modeling environment allows the test engineer to model a specific instance of an instrumentation system using the available algorithmic, signal, and hardware resources. This environment is a drastically simplified, menu-based interface that allows the test engineer to interact with the systems development process in a natural manner.

Model-Based Synthesis: Synthesizing Systems

Figure 2 represents the typical design flow for a model-based instrumentation system. The process will be described in general in this section, in order to give the overall structure of the process. Each of these phases will be described in detail in the following sections.

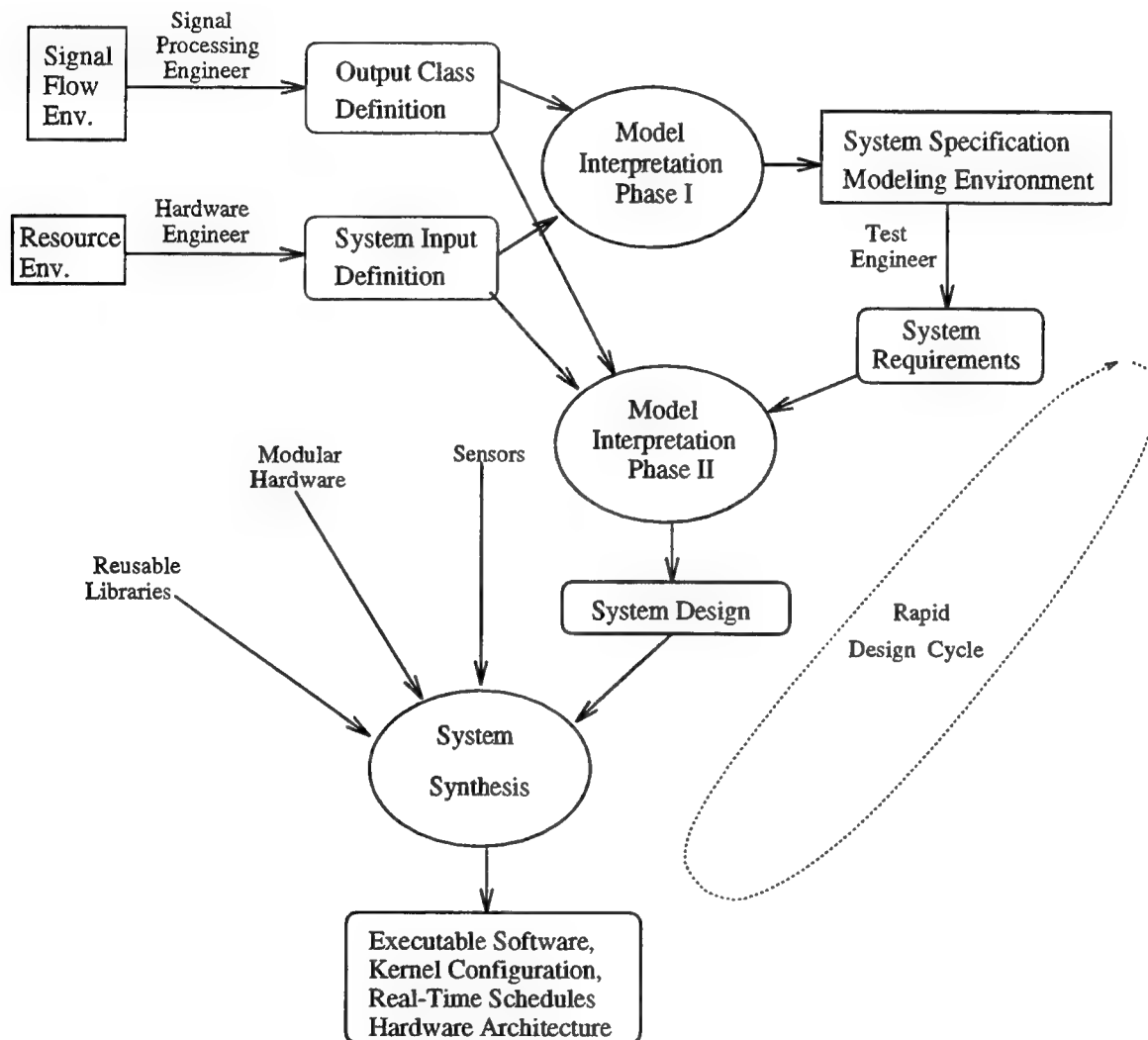


Figure 2: Real-Time Instrumentation System Design Cycle

In phase I, the Instrumentation Systems Engineer defines the capabilities and system inputs, using the signal flow modeling environment. The results of this modeling effort are the Output Class Definition and the System Interface Definition. The Output Class Definition process occurs

relatively infrequently. It is analogous to the generation of function libraries in standard software engineering practice. The relevant sets of system inputs, hardware resources, and system outputs are grouped together into a system structure model.

Phase I entails an interpretation of the component definitions referenced in the system structure model to create an environment for the Test Engineer, the System Requirements Specification Environment. The creation of this environment is analogous to compiling and documenting the function libraries.

The remainder of the design process occurs often: i.e. every time the system requirements change. During phase II, the System Requirements Specification Environment is used by the test engineer to create a requirements model database. This database is interpreted along with the output class database to generate a system software design, using a high-level system specification formalism.

In phase III, the system design specification is used, along with the hardware definition objects referenced in the system structure model, to synthesize a combined hardware/software architecture. The output is a hardware design, along with a scheduled, allocated, executable network of software modules, for the various processors. These outputs are used to build a parallel network and configure a modular kernel. The resultant structure then executes the program matching the requirements specified after Phase I.

The Phase II-Phase III process occurs rapidly, to meet changing user requirements. This cycle can also be used to support dynamically changing systems, with reinterpretation occurring online.

Application Example #1: The CADDMAS System

The CADDMAS consists of a loosely coupled set of processors, connected in an application-specific topology. This section will discuss the physical architecture of CADDMAS along with the functional distribution of tasks being executed on the processors.

Figure 3 show the overall CADDMAS architecture. The architecture is scalable to large numbers of channels and user interfaces. Each user interface has the same set of capabilities, and each channel is processed in a uniform manner.

The various processors in the system and their functions are:

- **User Interface:** The user interface hardware consists of a 386-or-better PC with a high

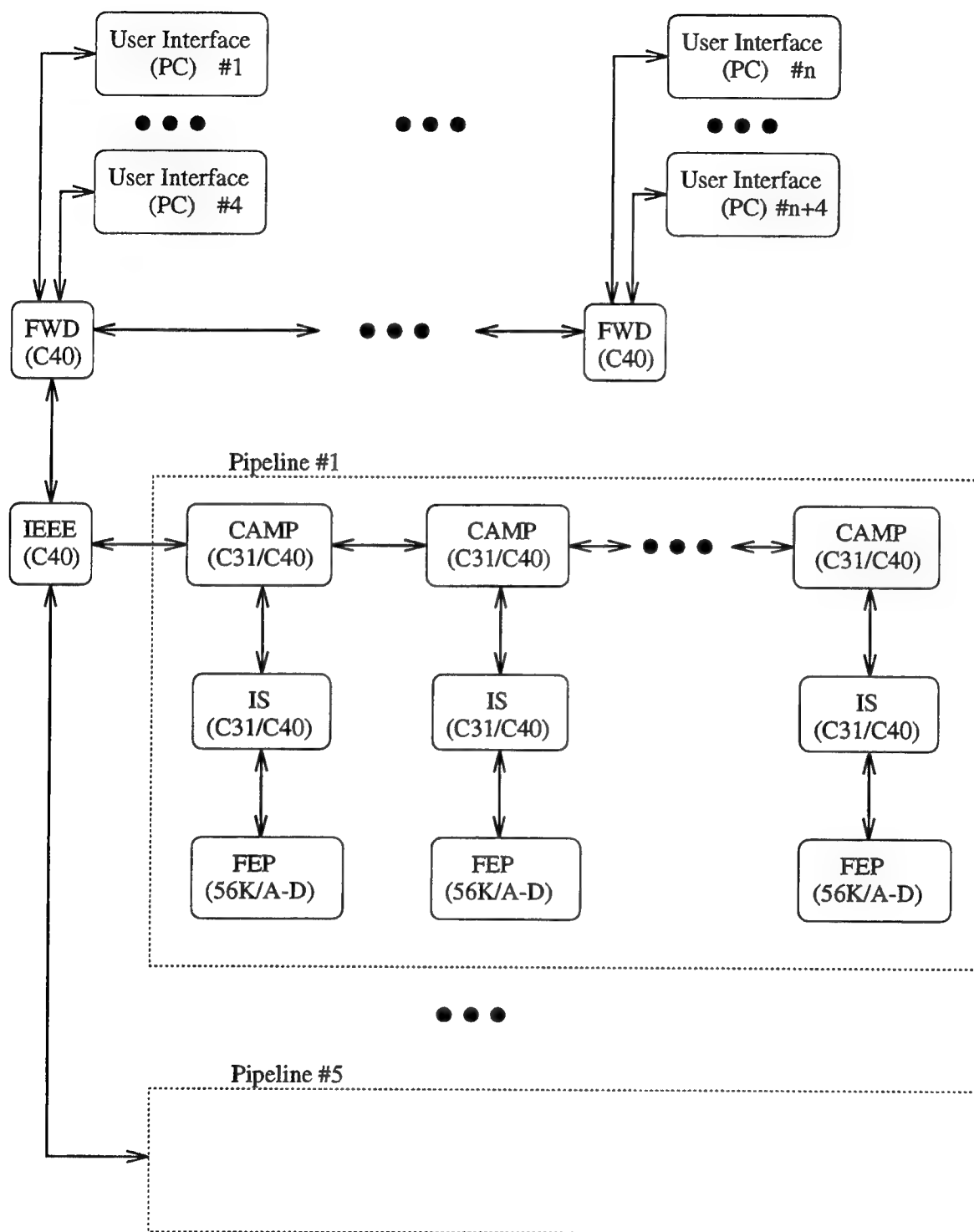


Figure 3: C40/C31 Architecture Diagram

resolution monitor, accelerated graphics card, keyboard, mouse, and optional printer. One of the user interfaces resets, bootstraps, and loads the network of processors with the proper executable software on system startup. All user interfaces execute a graphical user interface program with the following functions:

1. Accept user input to allow configuration of plot screens and selection of data sources.
2. Transmission of commands to the system of parallel processors to configure datapaths, request data, and control data processing modules, such as campbell and spectral envelope collection.
3. Acquire data from the system and maintain real-time updated copies.
4. Plot graphs as requested in the user interface configuration.
5. Print graphs to the local printer.
6. Store snapshots of data to the local disk.

The user interfaces also use the parallel system as a local user-interface network. Via this network, the user interfaces can update each other, with screen configurations, datapoint labels, and print/store commands. This facility allows a single user to control the operation of multiple user interfaces.

- **FWD:** The Forwarder serves as a data control and distribution network for the set of user interfaces. User interface commands propagate through these nodes to the network of processors, and are interpreted en-route. Tables of data requests are maintained for each user interface, allowing efficient routing of data packets. Data is acquired from the network via the IEEE processor and shipped to the proper set of user interfaces.

The data transmission function of these processors leaves the other CPUs lightly loaded. Thus, these processors are available for expansion of the CADDMAS algorithms.

- **IEEE:** The IEEE serves as a data routing processor, interfacing the network of User Interface/FWD processors to the network of data processing nodes. The data is merged from several processing pipelines. On a T800, 3 channels are available, while 5 are available on a C40 system. The commands from the user interfaces are distributed to each of the pipelines. The IEEE processor also converts data formats, since the C31/C40 processors use TI floating point format and the PC uses IEEE.

- **CAMP:** The Campbell processor performs the collection of spectral peak information into a database to be used for plotting campbell diagrams. This processor also gathers the data for the Engine Order and the RPM-Time tracking. In addition, the Campbell processor implements data forwarding up the pipeline towards the IEEE processor.
- **IS/IS_DB:** The IS processor (in the case of the C40) and the combination IS/IS_DB processors perform the following: **Engineering Unit (EU) Conversion, Complex Spectrum Computation, Spectral Magnitude/Envelope Computation, and Spectral Peak Detection.**

These computations are done directly on the C31 or C40 for TI-based systems. For transputer systems, the Zoran VSP performs the majority of the mathematical operations.

- **FEP:** The FEP samples the analog data at a frequency 2.56 times the system bandwidth and transmits out a link compatible with the processing nodes in the system.

Data Types

The information flowing throughout the system is characterized by a limited set of data types. These types are described below: **Raw Data:** The raw, integer data comes directly from the sampling system. **Time Data:** Time-domain data represents physical quantities converted into appropriate units. **Complex Spectral Data:** the complex spectral estimation resulting from a real-input Fast Fourier Transform(FFT). **Spectral Magnitude Data** is the absolute value of the complex spectral data. **Spectral Envelope Data** is the peak-hold envelope of the spectral magnitude data. **Spectral Peaks Data** is the peaks of the spectral magnitude data. **Campbell Data** is a two-dimensional database of spectral magnitude peaks, spectral phases, and correlated parameters collected over time within a range specified by the user. The data is collected, collated, and filtered over an entire test point. The driving parameter is commonly engine RPM, but time or any other parameter is valid. **Engine Order Data:** is a one-dimensional database corresponding to the peak spectral magnitude at a specific multiple of engine RPM (or any other measured parameter). The independent variable is the RPM/parameter. **RPM-Time Tracking Data** is similar to the Engine Order data, except using the maximal time-domain peak as a function of parameter.

CADDMAS Application Areas

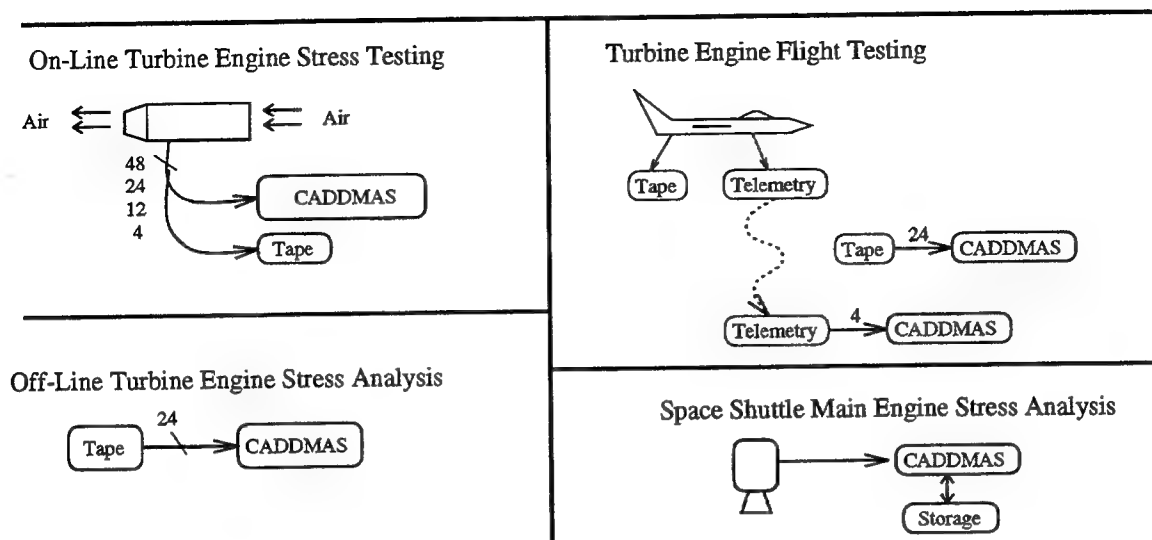


Figure 4: Applications of CADDMAS

CADDMAS Systems in Operation

The CADDMAS architecture has been employed in several systems (see figure 4). Both major engine test facilities have large permanent systems installed. The first facility uses a 64 processor, 2 GFLOP CADDMAS system based on the TMS320C40 and TMS320C31 processors. 5 interactive user interfaces are supported concurrently, processing 48 channels of data. The second system is based upon the Inmos T800, using approximately 36 processors. Several other systems are in operation: A flight test facility is using a small 12 channel CADDMAS system for online analysis of flight data. An extended CADDMAS is under development for NASA for use on the Space Shuttle Main Engine. This system incorporates advanced algorithms for vibration analysis.

Application Example #2: The DATVAL System

The steady-state data validation process at AEDC involves multi-step data acquisition and processing, that must be performed before the results of the validation process can be shown. Data is collected from sensors placed on a test article and is sent to a large (approximately 50,000 lines of source code) data reduction program. Output data from the data reduction program can then

DatVal Data Flow

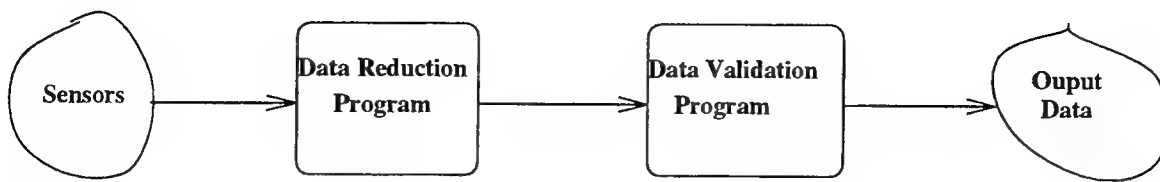


Figure 5: Data validation data flow

be fed to the actual data validation program (See figure 5). Currently, the data validation program lacks several features; no graphical user interface, no diagnostics, and limited visualization of the data are some of areas that need to be upgraded.

Data is sampled from the engine in blocks known as data points. Each data point contains several steady-state measurements from the engine. These data points arrive approximately every two minutes, but can arrive in much shorter intervals. Each data point must be processed to ensure the quality of the data before it is stored on magnetic tape, for delivery to the engine manufacturer.

A new data validation package (DATVAL), which is currently being brought on-line, uses a graphical interface and many added data viewing features to help the test engineer validate test data. Some of the new features include the ability to look at the data in its reduced form, the ability to view the data in the internal formats (with further reduction/computation performed) of the data validation package, tolerances for key data sets can be modified on-line by the test engineer, and rudimentary diagnostics can be performed. One of the key advantages of this new system is the ability to look at any and all data at any given time. With this new system, all of the functionality of the old system is maintained, and these several features only add to the usefulness of the system.

DatEdit, the DATVAL system editor, allows modification of the system on-line. This is the model editor (and run-time system synthesizer) used for the DATVAL system. It differs significantly from the model editor used for the CADDMAS system in several minute ways. Using this package, a test engineer can make changes to the system and, in a very short period of time, see the results of those changes. DatEdit uses the original data validation routines as part of the system model used to generate the run-time system. By easily allowing changes and automatically generating the DatVal system (run-time system) from the system models, DatEdit allows the test engineer great flexibility in the data validation process. If the engineer has a need to add more tests to the data validation system, he can add these new features with DatEdit during a test period.

Then, before the next data point arrives, these changes will be in effect.

Using model based techniques allows an engineer who is not trained in computer systems to build a custom, complex application in a very short amount of time. One of the goals of the DATVAL project is to allow a test engineer to customize the data validation process to enhance his ability to ensure the data is correct. DATVAL is a diversion from the other model based systems in one key area: the use of legacy code. By requiring the use of the original data validation routines, a new method for dealing with the model information contained in these routines had to be devised.

One of the problems with the DATVAL is the first-time system configuration. DATVAL uses the old data validation routines internally, adding a "wrapper" of functionality around the old routines. While building a system that uses these old routines is not very difficult, to take full advantage of the additional features of DATVAL additional information must be supplied. The set of internal variables and input variables that influence each check must be specified. Key data items such as tolerances must be specified separately, so as to allow the on-line modification of these items. Currently, this information must be entered manually.

A procedure has been developed that can extract most of this information automatically. This information is included in the model of the engine. Unfortunately, only the source code contains the model information. By extracting the data dependency information from the source code, the engine model data can be derived. A data dependency analysis will return a tree type structure that maps all of the inputs of the data validation program to their corresponding outputs. This information can be used for automatic system configuration.

A data dependency analysis tool (DDAT) has been developed. Currently, this tool can determine approximately 90% of the data dependency information contained in the data validation routines. Some small changes are needed to increase the effectiveness of this tool. After these changes have been made, testing can commence on the data reduction program. Work on the data reduction program has not commenced due to the large size of the program. Until the DDAT has been shown to effectively analyze the data validation routines, it will not be used on the data reduction program.

DDAT is a multipass analyzer. First, a package called Sage++ is used to parse the fortran source code into an object database used by DDAT. Several passes through the object database then occur. The first pass compiles all the common block variables into specific data structures. Next, the mapping of the input variables to the common block variables is extracted. Then, the mapping from internal variables to outputs is extracted. This allows a tree to be constructed that maps from the reduced data to the data error displays. When the data reduction program can be

analyzed, a complete mapping of outputs to sensors can be produced.

By using the DDAT, the DATVAL configuration will be much easier and less labor intensive, allowing the DATVAL package to be used much more efficiently. Currently, the DDAT is undergoing testing and modification. Modifications are being made to correct shortcomings in the algorithm that is used to extract the data dependencies. This algorithm is constantly changing to incorporate new features that improve upon the usefulness of DATVAL.

Application Example #3: The Motor Monitoring/Diagnosis System

The motor monitoring/diagnosis system uses the model-based architecture to represent the structure of the system under diagnosis. The modeling paradigm captures the physical and electrical structure of the motor and controller. From the electrical structure, a partial diagnosis approach can be derived, based on component functions and connection topology. Additional diagnostics information is to be modeled within the environment, to represent other physical interactions and accumulated experience from previous system diagnosis exercises.

From these models, a real-time diagnosis system is to be derived. (See figure 6). This diagnostics system interfaces directly to sensors on the motor, compressor, and controller. Timed readings of these sensors form the basis for the diagnostics process.

Additional predictive diagnosis processes are to be developed by watching trending information. For example, the trends of the current-voltage motor startup curves may reveal gradual changes in system performance. Analysis of current traces for insulator breakdown can also predict incipient insulator breakdown and subsequent coil damage. These are a small sampling of the analysis possible. The model-based approach captures the knowledge about the physical plant necessary to rapidly implement various diagnostics systems.

Acknowledgments

The authors acknowledge the kind support of the United States Air Force in making possible the work described in this paper as well as the CADDMAS Team in addition to the authors:

- James Mitchell: Arnold Engineering Development Center, Directorate of Technology.

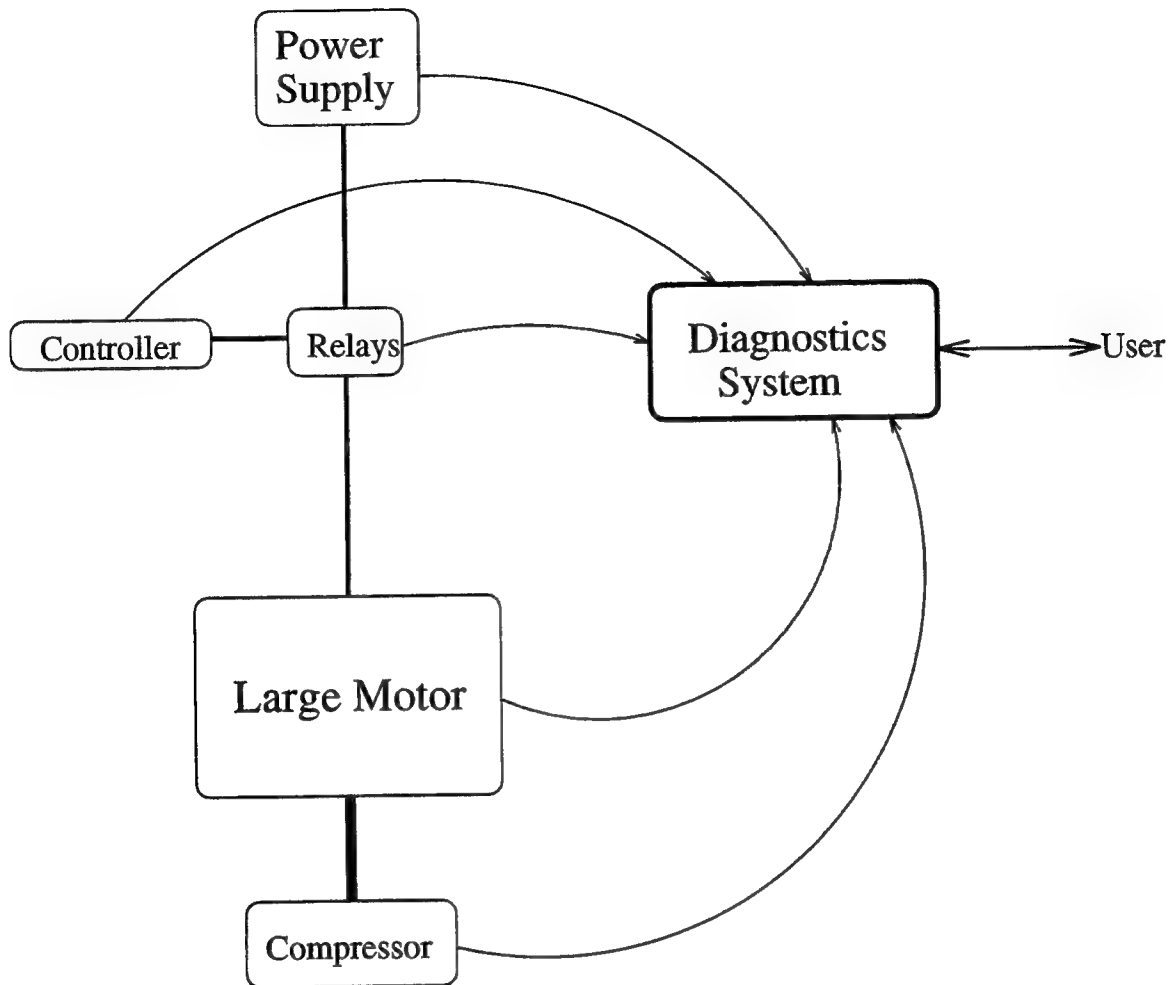


Figure 6: Simplified Motor System Diagnosis Architecture

- Tom Tibbals, Terry Hayes: Sverdrup Technology, AEDC Division
- Dr. Bruce Bomar: The University of Tennessee Space Institute

References

- [1] Bapty, T.: "Model-Based Synthesis of Real-Time Systems" Ph.D. Thesis Dept. of Electrical Engineering, Vanderbilt University, Nashville, TN., Dec. 1995.
- [2] Abbott, B, Bapty T, Biegl, C, Karsai, G., Sztipanovits, J. "Model-Based Approach for Software Synthesis", IEEE Software, pp. 42-53, May, 1993

- [3] Karsai, G., Sztipanovits, J. "A Visual Programming Environment for Domain-Specific Model-based Programming.", IEEE Computer, May 1995.
- [4] Ledecz, A., Bapty, T., Karsai, G., Sztipanovits, J., "Australian Computer Journal ", In Press.
Sztipanovits, J., Abbott, B., Bapty, T., Misra, A., "Model-Based Synthesis of Complex Embedded Systems", Proc. of the 1994 Complex Systems Engineering Synthesis and Assessment Technology Workshop, Washington D.C. July 19-20, 1994
- [5] Bapty, T.A., Abbott, B.A., Biegl, C. Ledecz, A. "Parallel Turbine Engine Instrumentation System", Proc. of the 9th AIAA Conference on Computing in Aerospace, San Diego, CA, October, 1993
- [6] Ledecz, A. Abbott, B. Bapty, T., Biegl. C., "Parallel DSP System Integration", Microprocessors and Microsystems, Vol 17, No. 8. 1993.
- [7] "Research on Intelligent Process Control Systems" Dept. of Electrical Engineering, Vanderbilt University, Technical Report #88-003, 1988.
- [8] F. Bodin, P. Beckman, D. Gannon, J. Gotwals, S. Narayana, S. Srinivas, B. Winnicka, *Sage++: An Object Oriented Toolkit and Class Library for Building Fortran and C++ Restructuring Tools*, Proc. Oonski '94, Oregon, 1994.

**TOWARDS THE REAL-TIME EXECUTION OF
TURBINE ENGINE SIMULATION PROGRAMS**

**Csaba Biegl
Research Assistant Professor
Vanderbilt University
Department of Electrical Engineering
Box 1840 Station B
Nashville, TN 37235
(615) 343 - 8197**

**Final Report for:
Summer Faculty Research Program
Arnold Engineering Development Center**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, DC**

and

Arnold Engineering Development Center

September 1995

TOWARDS THE REAL-TIME EXECUTION OF TURBINE ENGINE SIMULATION PROGRAMS

Csaba Biegl
Research Assistant Professor
Vanderbilt University
Department of Electrical Engineering

Abstract

The Arnold Engineering Development Center (AEDC) uses a variety of turbine engine simulation programs to enhance its testing capabilities. Traditionally, these models have been executed off-line, sometimes at processing speeds which are several orders of magnitude slower than the speed of the physical processes they model. Increasing demands for faster test turnaround times and data accuracy necessitate the integration of on-line simulation into the testing process. This paper describes two efforts towards this goal: a graphical user interface which is intended to speed up user interaction with the models, and a parallel processing toolkit used to convert some of the existing AEDC models for parallel execution.

TOWARDS THE REAL-TIME EXECUTION OF TURBINE ENGINE SIMULATION PROGRAMS

Csaba Biegl

Introduction

An important part of the testing of turbine engines at AEDC is the application of engine models for data validation. Turbine engine models are numerical simulation programs that model the behavior of the engine or some of its components under various operating conditions. Some models are based on exact modeling of the physical (transport, thermal and mechanical) processes inside certain engine components. Other models are less detailed, component-level simulations which try to approximate overall engine parameters from the engine layout and component characteristics.

Turbine engine models can be used for several purposes like design analysis, modeling the interactions between the airframe and the engine, education, etc.. From the point of view of an engine test facility, like AEDC, the most beneficial use of the models is to aid the testing operations. This requires models that are fairly accurate and have been verified in different operating modes of the engine. The development of accurate engine models is a time-consuming process that uses several possible input data sources:

- Data, component characteristics and sometimes incomplete models from the engine manufacturer
- Parameters inferred based on the geometry (duct diameters, etc..) and operating parameters (RPM, etc..) of various engine components
- Measurement data obtained from engine tests.

Typically, at the beginning of a test sequence, the amount of available data does not allow the creation of an accurate engine model. After test runs, some of the measured test parameters are used to verify and to refine the accuracy of the models. Simultaneously, the model is also used to determine some other engine parameters that cannot be measured in the test configuration. Thus the collection of test data and the refinement of the model are two interacting activities which proceed in parallel during the course of a typical engine test sequence.

Once the model has been refined to an acceptable accuracy level it can also be used to validate the test data. A turbine engine test facility is a very complex industrial plant (wind tunnels, compressors, extensively instrumented test cells, etc..) which must be highly reconfigurable in order to support various engine types and operating conditions. The probability of a component failure or a misrouted connection in such a plant is quite high. Most of the time these plant failures will manifest in invalid measurement data. Since running the engine test facility is quite expensive, it is imperative that such plant failures be

detected and corrected as early as possible. Accurate engine models play a very important role in the validation of the test data.

To best support the testing operations the engine model has to be available on-line during the test periods. Ideally, the simulation speed should be real-time or even super real-time so that events in the engine can be analyzed (or even predicted) as they happen. Most currently used engine models do not meet this requirement. On current single processor computers their execution speed is two or more orders of magnitude slower than the real physical process they simulate. This, of course, also depends on the computer platform being used. Future advances in hardware and software technology may narrow the gap. Alternatively, most engine models could be adapted for parallel execution.

In most test scenarios the models can be quite useful even if they cannot execute in real-time. Typical engine tests proceed with alternating periods of relatively short (few seconds to minutes) engine maneuvers (acceleration, deceleration, throttle snap, etc..) and longer (several ten minutes to few hours) setup periods for the next test point. The engine models can be used in the setup periods to analyze and validate the data from the previous maneuver. Many engine models execute fast enough for this kind of usage on today's high performance workstations.

Adequate execution speed of the model is just one requirement for it to be usable in on-line testing. It is also necessary that the simulation be able to interact with the user efficiently. This includes user friendly model input controls and simulation result presentation. These requirements are best satisfied with a well-designed integrated graphical user interface for the engine model.

Overview of modeling techniques

AEDC currently uses different types of turbine engine or engine component simulation codes. Many of these are potential candidates for on-line execution in the test environment, provided that the data interfacing and execution speed issues can be resolved.

- Generic engine models (ATEST): ATEST models are built from pre-coded engine component model blocks, like models for compressors, burners, turbines, ducting, etc.. These blocks model the overall characteristics of the given component, they are not detailed models of the physical processes inside the component. A complete engine model is built by specifying the components, their parameters (for example compressor map tables, sizes, etc..) and interactions. A numerical solver module is used to match the various component input/output parameters until the whole model converges to the specified operating point. ATEST can be used to perform either steady state or dynamic simulations. Sometimes the ATEST program is used as part of a more complex simulation system.
- Detailed component models (DYNTECC, ATEC, etc.): These simulations model the actual physical processes inside a single engine component. For example, the DYNTECC program models the processes inside the stages of a turbine engine compressor. It divides the compressor volume along a

one dimensional grid into up to 60 axial segments and calculates mass, flow, energy, etc. balances for each of these segments. It also allows the segmentation of the compressor volume into up to eight circumferential segments (each of which can again have up to sixty axial segments).

- Three dimensional grid models (TEACC): These models use the techniques of Computational Fluid Dynamics (CFD) to even more accurately the processes inside various engine components. TEACC models turbine engine compressors by dividing their volume along a three dimensional grid, which allows much more accurate simulation. Similarly to other CFD codes, TEACC is very computationally demanding. For this model category execution on a supercomputer or parallel machine is necessary even in off-line use.

Software engineering aspects of the current modeling techniques

Almost all of AEDC's models are written in the FORTRAN language. Originally none of them was intended either for parallel execution or for integration with a graphical user interface. There are several reasons for the use of the FORTRAN language:

- Efficiency: Although there are more modern programming languages available, FORTRAN is still one of the best languages for fast numerical computations. It is exactly the lack of "modern" features like pointers, etc.. which makes it relatively easy to write very good optimizing compilers for it. Additionally, the emerging High Performance Parallel FORTRAN (HPF) standard promises a relatively easy way to create parallel FORTRAN programs -- at least for certain kinds of parallel architectures and for certain types of problems.
- Historical reasons: most of the existing engine codes were developed on mainframes and later ported to PC-s and workstations. On the mainframe the natural development language was FORTRAN. Later the universal availability of the FORTRAN language made the porting easier.
- Developers: Most engine models are written by aeromechanical engineers who want to concentrate on the aeromechanical aspects of the problem instead of software engineering issues. Thus the relative simplicity of FORTRAN is again an advantage.
- Models are developed based on existing code libraries for simulating various components, component interactions or physical processes inside components. The typical development process involves writing a few engine dependent subroutines and initializing various constants, tables, etc.. with the parameters of the engine. The thus customized FORTRAN library becomes the model of the given engine. In practice there are various versions of the same code library customized for different engines that diverge during the lifetime of the model as changes are implemented.

The input and output of the models is typically done with formatted FORTRAN data files. Most models use the FORTRAN "namelist" construct for parameter input. Output files are mostly just formatted tables

of numbers that have to be post-processed with stand-alone plotting programs to generate graphical output.

Graphical User Interface Design and Implementation

Considering the background information presented above the design of a graphical user interface (GUI) for the existing FORTRAN engine models must satisfy the following requirements:

- **Features:** The main goal of the user interface is to speed up the interaction between the human user and the engine simulation program. This means that the GUI must provide an easy to understand display of simulation data (plots) and it must provide a straightforward interface to control the simulation program. Additionally, in some applications it is desirable that the simulation be interfaced to the on-line engine test data system for providing direct input of measurement data into the model.
- **Universal applicability:** due to the large number of engine models and their variants the user interface must be easily adaptable to any FORTRAN model. Preferably this adaptation should involve only FORTRAN programming because this way the developers of the original models could do it.
- **Portability:** the user interface should be portable to PC-s running DOS and Windows, and to workstations running UNIX and the X Window system

GUI development history; The DYNTECC User Interface

The graphical user interface for the DYNTECC program was the first interface developed for AEDC engine models. The interface supports one-dimensional plots of time history data or parameter versus parameter plots. Plots can be created from FORTRAN output files produced during previous runs or interactively, using the DYNTECC code linked into the interface. In the second case the plots are updated on-line as the simulation progresses. The interface supports four on-screen plot regions, each of which can be configured to display any of the available simulation parameters.

The user interface has hard copy capabilities as well. Individual plot regions can be printed on a variety of supported printer types (several laser, ink jet and dot matrix printers), or a dump of the whole user interface screen can be made into a graphics bitmap file (PCX format). The screen dump can later be included into documents or presentations.

To control program inputs, the interface supports the editing of the various FORTRAN namelists that make up the input file DYNTECC uses to control the simulation. This is done using a hierarchy of menus, the first of which selects the namelist to edit, and the second selects the edited field. On-line help is available for each of the menu choices.

The DYNTECC user interface has been implemented using SUIT, a portable user interface toolkit developed at the University of Virginia. The original DYNTECC FORTRAN code is compiled

using F2C when linked with the interface. (Command line versions of DYNTECC are still compiled with the original, native FORTRAN compiler.) The interface is available, and has been used, on several platforms, including IBM PC-s running both DOS and Linux and Hewlett Packard workstations.

The ATEST/SCIDNT User Interface

The second graphical interface was developed for a modified version of the ATEST engine modeling code. This ATEST version was integrated with the SCIDNT parameter identification package. The goal of this project was to develop a method for refining ATEST models by adapting a few key engine model parameters based on simulation run results and selected measurement data records. This second generation user interface had similar services as the DYNTECC user interface described above, but with a few important additions:

- Parameter classes: Unlike the DYNTECC interface which operates with only one kind of data for every parameter (i.e. the current simulation results), the ATEST/SCIDNT interface supports several kinds of values for every parameter. This is necessitated by the adaptation mechanism, since the user may want to examine plots of the same parameter from several sources including measurement data, unadapted and adapted model runs. Additionally to the three parameter classes mentioned above, two more classes are supported: design point values and adaptation histories. Design points are nominal values of certain parameters that are available in the ATEST model. Sometimes their inclusion in the plots can serve as a reference point. Adaptation histories are records of engine model parameter during the SCIDNT adaptation run. They can be used to analyze the progress of the adaptation process and possibly select more suitable adaptation parameters or gain factors.
- Multiple plots per plot region: The ATEST/SCIDNT interface supports plots of multiple parameters or different records of the same parameter in the same plot region. Each parameter will be plotted against the same independent variable (the default is time), but the parameter class of the independent variable will be matched against the class of the variable on the Y axis. For example, if parameters "P1" and "P2" are plotted versus time in the same region, and "P1" is, say, a measured variable while "P2" is a simulated variable, then the X axis will be scaled in time, and the plotted time interval will be the union of the available simulated and measured time points.
- Handling of custom FORTRAN file formats: unlike the DYNTECC interface which, when plotting from results of previous runs stored in files, can input only raw FORTRAN output data (i.e., a table of numbers without page, column or row headings), the ATEST/SCIDNT interface can read formatted FORTRAN data. The interface incorporates a set of customizable file scanner library routines that are capable of parsing various formatted FORTRAN output files. This code can also extract parameter names from the files while skipping page headers and other irrelevant information. Adapting this

library to a new FORTRAN data file format typically requires writing a short (10 .. 20 line) subroutine.

- Minor improvements: these include support for different screen configurations (instead of the DYNTECC interface's fixed four window configuration, the user can select configurations with one to sixteen plot windows), a simulation screen window to display the messages originally printed by the FORTRAN model on the console and the capability to run external (i.e. not linked into the interface) models and then scan the simulation results from the produced data files.

The DYNTECC and ATEST/SCIDNT interfaces were completed during the 1993 and 1994 AFOSR Summer Research Programs, respectively. The design and implementation experience gained with them was utilized in the universal simulation user interface that was completed during this year's program.

The Universal Simulation User Interface

Based on the preliminary DYNTECC and ATEST/SCIDNT work a new graphical user interface library for turbine engine modeling was designed whose main goal is to provide universal support for converting existing FORTRAN models. This interface reuses many of the components from the ATEST user interface. The following new features and improvements were added:

- Support for multiple plot pages: In typical applications even the ATEST interface's up to 16 plot windows were found to be inadequate, especially when the simulation generated many parameters. The universal interface introduces the concept of plot pages, each of which can hold a user settable number of plot windows.
- Support for user-written data plotting code: Many times the simple parameter versus time and parameter versus parameter plots are not sufficient to best visualize the processes modeled by the simulation code. The universal interface supports user written plot subroutines that can be used to provide extended plotting capabilities. These user registered routines can either generate plots directly (using the interface's plot routines) or can be used to transform data records generated by the simulation. A variety of "standard" data transformation routines (add, subtract, multiple, derivate, integrate, etc...) have been included in the interface.
- Unrestricted parameter classes: The ATEST/SCIDNT interface's idea of supporting multiple data sets for the simulation parameters was extended in a way that permits the storage and handling of an unlimited number of data classes. (Unlike the ATEST interface in which both the number and the function of the parameter classes were fixed.)
- Support for run-time controls for the simulation. The interface allows the programmer to create and read or set the value of control widgets for various simulation parameters.
- Modularization of the code: The new interface can be used for data visualization as a stand-alone program. Customizing it for any application can be done from user code (typically the MAIN

program) simply by "registering" various subroutines into the interface. These user registered code pieces include simulation codes, data file scanners for various custom file formats, plot data conversion functions and plot display subroutines.

The main software components of the new interface design, which can be seen in Figure 1., are:

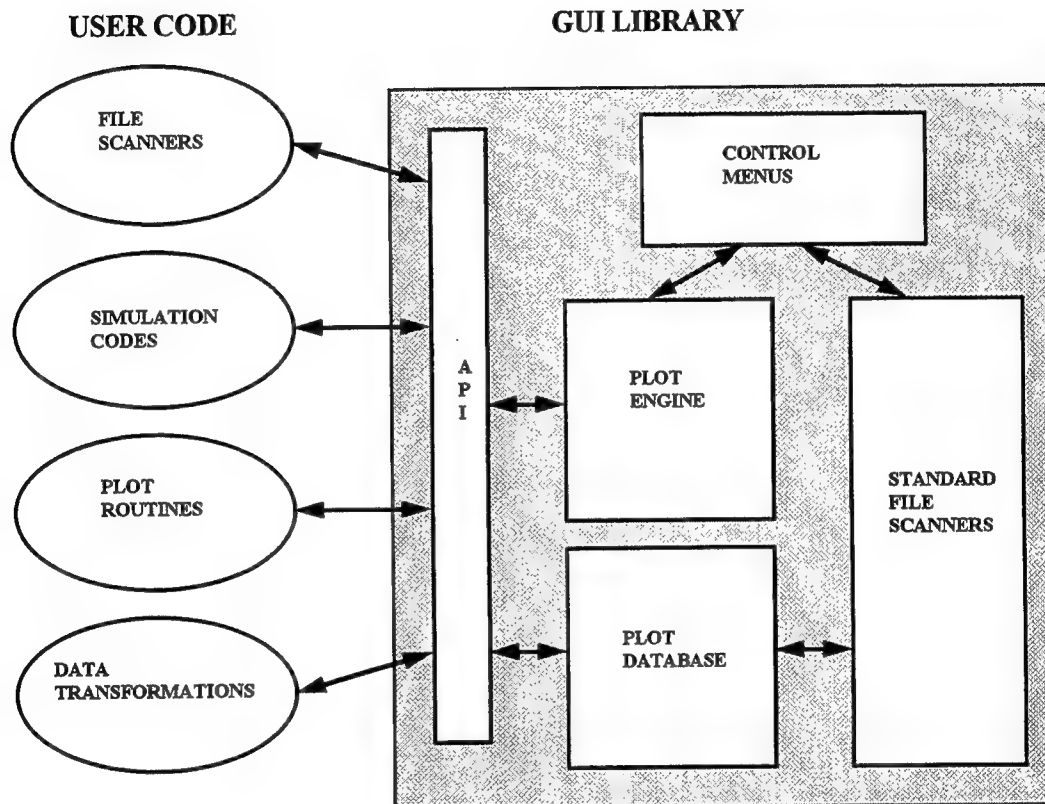


Figure 1. GUI Architecture

- **Plot Database:** This data base contains records of simulation variables generated either by loading data files or by running the linked in simulation code (if any). The data base can be searched by parameter names and classes. Each parameter can have several data records associated with it that are keyed by class. The Plot Database also allows user registered simulation subroutines to enter data during a simulation run. In this mode plots are updated on-line as the simulation progresses.
- **Plot Engine:** This component is responsible for generating the plots that are displayed in the plot windows of the user interface. A slightly modified version of the public domain GNUPLOT plotting program was used for this purpose. The Plot Engine operates in either of two modes: it can plot data from the Plot Database or display data generated by a user registered plot subroutine.

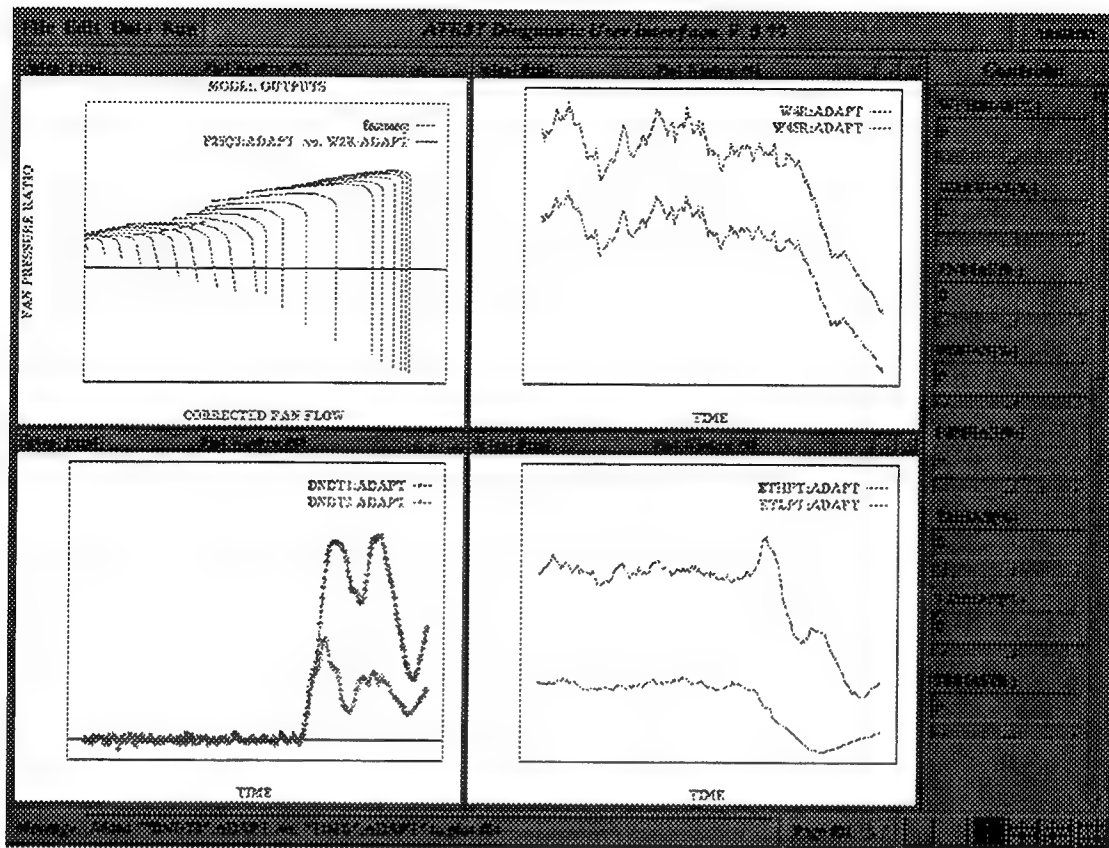


Figure 2. Example ATEST Application

- **Standard Data File Scanners:** The interface provides the capability to scan a few "generic" data file formats. These include binary data files and horizontally or vertically tabulated formatted tables with or without parameter name headings. (The ASCII outputs from most popular spreadsheet programs can be scanned this way.) For more customized file formats it is possible to register routines for the given file layout using the services of the Scanner Library.
- **User Interface Menus and Controls:** these contain the menu options through which the various screens, panels and menus of the interface are accessed. These control components are responsible for coordinating the operations of the other interface services. They select the plot window layout, assign parameters from the Plot Database to the various plot windows for display, configure the plot windows, etc.. Most of the control services can be invoked via the top-level pull-down menu of the interface.
- **Application Program Interface (API):** this layer contains the entry points through which user code can be registered into the user interface and the service routines usable by these user routines. The API is currently provided in two versions: for C or C++ programs and for FORTRAN programs compiled with the AT&T F2C translator. The FORTRAN entry points are implemented as a wrapper around the C entry points, this way it is to adapt the interface to a different FORTRAN environment.

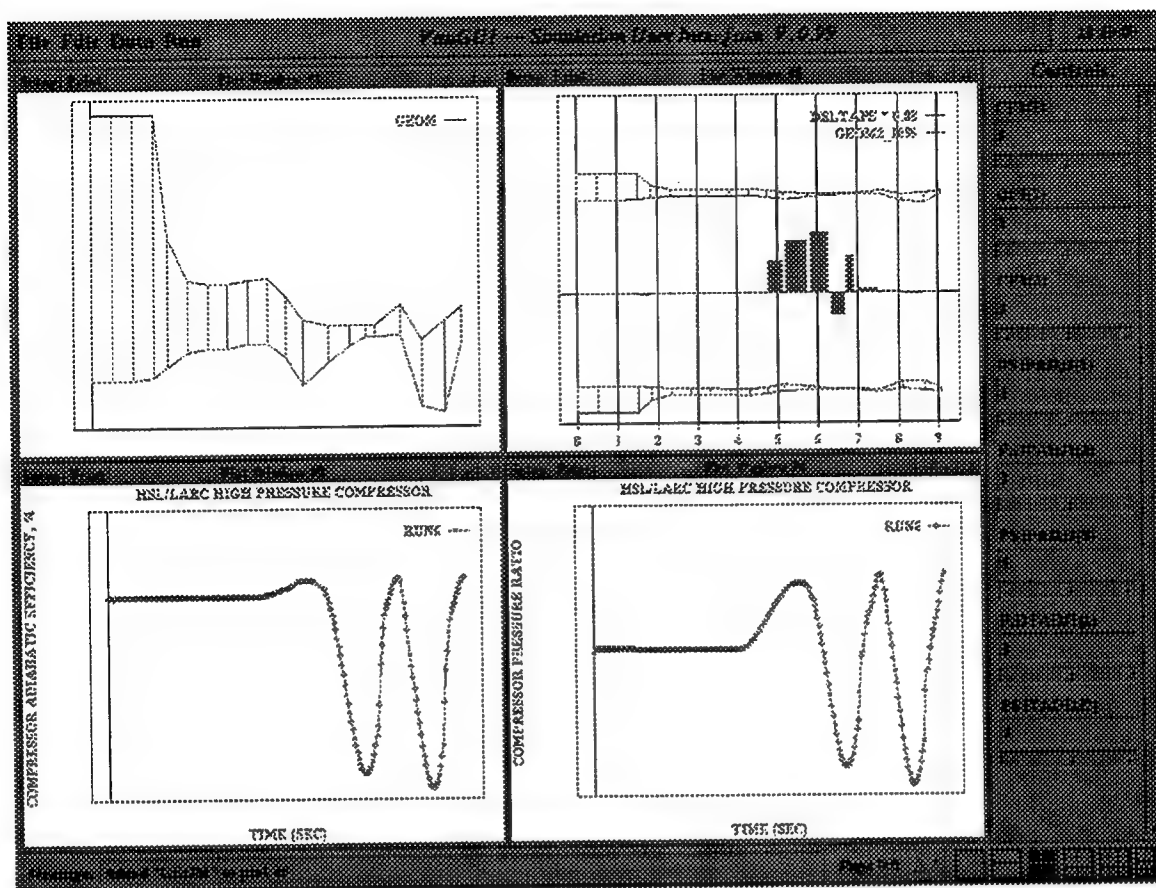


Figure 3. Example DYNTTECC Application

The new universal interface library has been implemented for a variety of platforms, including IBM PC-s running DOS (full screen) or Windows (requires a 32 bit Windows version, like WIN32S, Windows 95 or Windows NT) and several UNIX workstations with the X window environment (Linux, SGI, HP),

The universal interface was written in C++, using interface elements (buttons, scrollbars, etc..) which were adapted from public domain toolkits (so called "widget" sets) or implemented from scratch when necessary. Writing new widgets was necessary because none of the commercially available "portable" toolkits cover all of the targeted platforms (DOS graphics, Windows and the X Window system) -- at least not without unacceptable restrictions (like limiting the compiler choices under DOS and Windows or requiring a Motif license for workstations).

The interface has been successfully applied in several stand alone applications to visualize data generated by various AEDC codes. Additionally, both the DYNTTECC and ATEST programs have been interfaced with it. Example output screens from the ATEST and DYNTTECC applications can be seen in Figures 2 and 3, respectively.

Parallelization of FORTRAN engine models

Besides speeding up user interaction, the other necessary precondition of using the existing AEDC engine models on-line in a test environment is sufficient execution speed of these models. Work has begun on this issue during this year's Summer Research Program. Distributed workstation networks have been chosen as the target architecture for this work. The reason for this is their better availability, more flexible applicability (workstations can be used independently by several users during periods when no testing is performed), affordability and the inherently better scalability of distributed architectures. Many workstation vendors also offer high speed interconnection networks for their products, which can significantly improve the computing performance versus communication bandwidth ratio of these architectures. An example of these systems is the IBM SP-2 computer.

To provide a development environment for this work, an eight processor parallel distributed machine was built during the Summer Program. This machine consists of eight Pentium® PC clones interconnected via a high speed (100 Mbits/sec) Ethernet network. Every processor runs a copy of the Linux operating system, which is a public domain UNIX clone. The parallel machine also includes the necessary system management tools which is a relatively small collection of shell scripts added to the standard Linux distribution to facilitate the control of the distributed processors.

Automatic parallelization of FORTRAN code

The next issue after configuring the hardware platform was to select the methods and evaluate the tools that are available for converting the existing AEDC turbine model codes for parallel execution. Generating parallel code from FORTRAN input is a technique that has been used with success for some time. FORTRAN is a primitive language when compared to later languages (ADA, C, Pascal, etc.). This simplicity is in fact an advantage when creating parallelizing compilers. In modern languages which support pointers various subtle interactions are possible between variables — for this reason it is much harder to write a correct and efficient parallel code generator for them.

Unfortunately, the most successful applications of this technique are the FORTRAN compilers for shared memory MIMD or SIMD architectures. One of the best examples is the vectorizing FORTRAN compiler for the Cray supercomputers. These compilers work by parallelizing on the instruction level. They use techniques such as unrolling loops and assigning the iterations of the loop to different processors, etc.. These techniques work well in shared memory architectures, but in distributed systems their applicability is limited. As shared memory parallel computers are not scalable due to memory contention problems, this effectively limits the possible speedup which can be achieved using these techniques.

Generating good parallel code for the better scalable distributed parallel architectures is a more complex task. On these architectures the relatively simple instruction-level parallelizing does not work very well due to communication overhead problems. The desirable way to parallelize on these

architectures is to find larger blocks of code which can be executed in parallel before interaction with other processors is necessary. This requires a far more sophisticated program dependency analyzer than in the case of instruction-level parallelizing compilers. Additionally, the code generator is more complex as well, since it has to generate calls for some underlying communication mechanism.

For these reasons parallelizing FORTRAN compilers for distributed architectures are just starting to emerge. Many times these compilers rely on the High Performance FORTRAN (HPF) extensions of the FORTRAN 90 language standard. HPF allows the programmer to declare how a data structure such as a large array should be distributed and aligned between different processors when the program is executed on a parallel architecture. The HPF compiler will attempt to distribute the data according to the declarations, place code that operates on the data on appropriate processors, and generate the necessary communication and/or synchronization code. This implies that part of the responsibility for efficient parallel code generation belongs to the programmer. It is possible to decompose a problem in HPF in a very inefficient way.

Two typical parallel code generators which claim FORTRAN support on distributed architectures are the SAGE toolkit and the FORGE family of compilers. The SAGE toolkit is being developed at the Department of Computer Science, Indiana University, Bloomington, under a DARPA grant. Its primary aim is to implement a parallel version of the C++ language (pC++ -- Parallel C++), but it supports C and FORTRAN as well. The program source is first translated by one of the available compiler front end translators (pC++, C or FORTRAN) into an intermediate form. The output of this phase is a (typically huge) dependency file that contains the representation of all of the program's instructions and their interactions. This dependency output in itself is a valuable resource, various browsers can be used to analyze program structure, calling sequences, data usage, etc.. The toolkit also provides code generators that can compile the dependency file for various shared memory and distributed parallel target architectures. The supported distributed target environments include the Intel Paragon system and two widely accepted message passing standards for distributed computing: PVM (Parallel Virtual Machine) and MPI (Message Passing Interface). The SAGE FORTRAN front end currently is able to process FORTRAN 90 input. Support for HPF extensions is being worked on. The current implementation will not parallelize instructions operating on global data. This means that older style FORTRAN IV and FORTRAN 77 programs will not be parallelized extensively as these rely mostly on global data in COMMON blocks.

The FORGE compiler family is a commercial product offered by Advanced Parallel Research, Inc. It contains both interactive program dependency analyzers/browsers and batch mode parallelizing compilers. The interactive tools only support FORTRAN 90, while the batch mode compilers support both FORTRAN 90 and FORTRAN 77. The compilers are available for most commonly used UNIX workstations. They generate FORTRAN 77 output for either shared memory or distributed architectures. The distributed architecture output contains calls to FORGE's own communication library. This library is

implemented as a wrapper on top of commonly available communication interfaces such as PVM, EXPRESS, TCP-IP, etc.. Most of the tools in the compiler family seem to be geared towards the FORTRAN 90 language and HPF. Only one of the batch mode compilers attempts to parallelize older style FORTRAN 77 programs. Judging from its manual the achievable degree of parallelism is not very high in this case.

In summary, both of these tools will generate better parallel code from FORTRAN 90 input with HPF directives than from FORTRAN 77 input. In fact, it seems like old style FORTRAN 77 programs which rely extensively on global variables can never be parallelized by automatic tools to the same extent as HPF input.

Explicit Algorithmic Parallelization

The other avenue for parallelizing existing FORTRAN code is to understand the algorithm being used and distribute the different functional blocks of the program explicitly to different processors. This also requires that explicit calls to an underlying communication library are made at the points where synchronization and/or data exchange is necessary. There are many communication libraries which can be used for this purpose (PVM, EXPRESS, etc..)

The advantage of this approach is that many times the achievable parallelism is higher. The disadvantage is that it requires changes to the original program which may necessitate maintaining two separate versions of the code. (Or at least to use conditional compilation — for example with a C preprocessor.) Also, the algorithm of the original program needs to be analyzed in order to find the best decomposition. The dependency analyzer components of some automatic tools could be used to aid this process. The following is an overview of possible explicit parallelization avenues for each of the AEDC simulation problem domain areas listed before:

- **DYNTECC:** The most obvious way to parallelize the DYNTECC code is to run each circumferential segment on a separate processor. As there is no interaction between the segments, the communication overhead in this case is very low. The parallel execution of the control volumes can be generalized as the parallel finite element simulation problem: A grid of points is defined, some algorithm has to be executed for every grid point which calculates the new values for the given point based on the point's and its neighbors' state. When running such a simulation on a distributed architecture, adjacent grid points which have been assigned to different processors have to exchange data after every iteration. The optimum assignment is influenced by factors like the dimensionality of the grid, the execution versus communication speed of the hardware, etc.. In the case of the DYNTECC code the communication overhead is very favorable since it uses a one dimensional grid, the amount of data to be exchanged between adjacent grid points is low (~10-20 floating point numbers), and a relatively involved calculation is performed for every grid point.

- TEACC: The same considerations should apply as for the DYNTECC code. It should be taken into account that TEACC uses a three dimensional grid. We have no information at this time about the complexity of the computations that are performed on the grid points.
- ATEST: The "traditional" way to parallelize a block oriented dynamic simulation is to divide the blocks among the processors, and propagate block input-outputs between the different processors. Unfortunately, in the case of the ATEST program the component blocks form a linear chain, so practically no parallelism can be achieved this way. An other possibility is to replace the solver with an algorithm that can utilize multiple, concurrently executing copies of the model in its search for convergence. In the case of the ATEST program it is also possible that ATEST itself does not need to be parallelized. Its speed when executed as a stand-alone simulation seems to be sufficient. When used as a subroutine from a larger program (e.g. a parameter optimization package) it may be better to parallelize the calling program.

Initial Parallelization Results

Work has started during the 1995 Summer Program to parallelize an application of the ATEST program for on-line data validation. In this application, developed in cooperation with AEDC personnel, the ATEST model executes concurrently with the experiment, and is used to verify the measurement data. This is done by establishing tolerances for the differences between measured and simulated parameters. Whenever these tolerances are exceeded, a diagnostics algorithm is triggered. This algorithm tries to "explain" the discrepancies by trying to fit a vector of offset values to the measured parameters. These offsets can model either engine component failure and/or drift, or sensor errors in the test cell.

The original ATEST model for the given engine on which this diagnosis was performed executed at about 20% of the real-time speed on a Pentium/100 platform. Whenever the diagnostics needed to be executed, the program slowed down considerably, as more calculations had to be performed for every input sample.

This code was parallelized for the eight Pentium platform using a pipelined approach. This way, successive input samples were assigned to different processors, and the results the merged into a single time sequence. This was made by the fact that the state variables of the ATEST models changed very little between successive time points. The differences between the outputs of the sequential and parallel code are typically less than $1.0e-05$. This is a very efficient way to parallelize this type of code, as very little interaction is needed between the different processors. Speedup results on the 8 processor Pentium system indicate that the converted code is capable of executing at real time speeds whenever the diagnostics algorithm is not triggered. Work remains to be done on the diagnostics algorithm, as the speed of the parallel version drops significantly when a discrepancy is detected.

Conclusions

The goal of the research described in this document was to enable the introduction of existing turbine engine modeling technologies into the test environment. Work has progressed on two fronts: developing a user interface for the models which is operable in an on-line test environment, and to speed up the simulations by executing them on parallel architectures. The user interface work is a continuation of a long term effort, by the end of the Summer Research Program an almost complete, extremely useful version has emerged. The parallelization work has just been started, so most of the work was geared towards establishing an environment, and to study the applicable methods. Nevertheless, one application, the ATEST diagnostics tool, shows encouraging initial results.

TIME-OF-FLIGHT MASS SPECTROMETER SAMPLING SYSTEM
FOR AEDC IMPULSE FACILITY

Frank G. Collins
Professor of Aerospace Engineering
Department of Mechanical and Aerospace Engineering

The University of Tennessee Space Institute
Tullahoma, TN 37388

Final Report for
Summer Faculty Research Program
Arnold Engineering Development Center

Sponsored by:
Air Force Office of Scientific Research
Boiling Air Force Base, DC

and

Arnold Engineering Development Center

August, 1995

TIME-OF-FLIGHT MASS SPECTROMETER SAMPLING SYSTEM
FOR AEDC IMPULSE FACILITY

Frank G. Collins
Professor of Aerospace Engineering
Department of Mechanical and Aerospace Engineering
The University of Tennessee Space Institute
Tullahoma, TN 37388

Abstract

Presently there is great interest in developing high enthalpy test facilities for improving our understanding of high speed atmospheric flight and developing accompanying propulsion systems. Typical test facilities, such as the AEDC Impulse Facility, have millisecond run times which are terminated upon arrival of the driver gas, usually helium. A means must be developed for directly measuring the precise moment when the helium arrives in the test section. This can be accomplished with a time-of-flight mass spectrometer. A portion of the test section flow must be sampled, with an isokinetic skimmer system, and fed into the mass spectrometer. The present work describes bench experiments with the mass spectrometer and computations that were performed on a preliminary skimmer system. The skimmer system consists of three co-axial skimmers. The first skimmer must swallow the shock wave and the second must greatly limit the amount of gas which proceeds to the third skimmer. The third skimmer must be in free-molecule flow and define a beam of molecules which enter the mass spectrometer. The flow through the last two skimmers was computed using the Monte Carlo technique.

TIME-OF-FLIGHT MASS SPECTROMETER SAMPLING SYSTEM FOR AEDC IMPULSE FACILITY

Frank G. Collins

Introduction

Recently there has been great interest in producing high-enthalpy hypersonic (hypervelocity - Ref. 1) flows. The interest has arisen from the examination of aerobraking for orbit transfer on earth and Mars, the development of single-stage-to-orbit vehicles, and the development of air-breathing propulsion systems such as scramjets. These applications lead to "real gas" effects when the flow is decelerated along body surfaces or when combustion occurs at high density within the propulsion elements. "Real-gas" effects include vibrational excitation, dissociation, possibly ionization, chemical reactions, complex surface interactions, and flows that are not in thermal or chemical equilibrium. The state of the flow cannot be considered to be a small perturbation from an equilibrium state. These "real gas" effects appear when the flight speed is greater than approximately 4km/s (Ref. 1). Because of their complexity, high enthalpy flows must be explored by a combination of ground testing, flight testing, and computational fluid dynamics (CFD). Ground testing can be used to increase physical understanding of high enthalpy flows, provide validation experiments for CFD codes, and test individual vehicle components and propulsion elements.

Ground-based facilities cannot fully simulate the free-stream conditions that hypervelocity vehicles will experience (Ref. 1). While ballistic ranges come close to matching the conditions, they have the drawbacks of requiring small models and low free-stream density, of having difficulty in maintaining a prescribed model attitude, and of possessing short testing times. Recently success has been obtained in shock tubes operated in the reflected shock mode. The free piston tunnel developed by Stalker (Ref. 2) can provide stream speeds in the range of 7-8 km/s, stagnation enthalpy up to 40 MJ/kg, and test times of 1-3 ms. The flow through the

nozzle can be partially dissociated, which contaminates the free stream flow because it does not simulate flight conditions, but this is not a serious problem at lower speeds, such as are produced in the AEDC Impulse Facility.

It is important to calibrate the free stream of a hypervelocity facility by direct measurements (Ref. 3). Reliance upon nozzle calibrations and the use of initial reservoir conditions is not recommended because of the unknown effects of viscous and radiation losses. In particular, it is desirable to measure the chemical composition of the stream with time to determine the duration of the test. The test is terminated when the driver gas reaches the test section.

The most direct method for measuring the chemical composition is with a mass spectrometer that samples a portion of the gas stream. The earliest use of a mass spectrometer for this purpose was by Crane and Stalker (Ref. 4). They used a quadrupole mass spectrometer which could measure the concentration of only a single species during each test. In contrast, the rapid means by which a time-of-flight mass spectrometer (TOF/MS) determines an entire spectra makes it ideal for making repeated measurements during the short duration of a piston driven shock tunnel (Skinner, Ref. 5). Laser-induced fluorescence (LIF) and the newer planar laser-induced fluorescence (PLIF) techniques can also be used to make species measurements during short duration tests by employing pulsed lasers. However, these methods have the disadvantages of having to tune the laser to a single species for each tunnel firing, in common with the quadrupole mass spectrometer, and of requiring optical access to the tunnel. PLIF does have the distinct advantage of providing spatially correlated, instantaneous measurements whereas the TOF/MS can only provide measurements at a single location (Ref. 3). These optical methods are presently in their infancy and require maturation before they can be used in test facilities. Comparison of optical and mass spectrometer measurements will speed this process. However, the mass spectrometer will always be required when optical access is not

possible, especially for measurements within nozzles and inlets.

The present work describes progress made toward the design and operation of a TOF/MS system for the AEDC Impulse Facility. Arrival of the helium driver gas will be determined using the Comstock Model TOF-101 TOF/MS. The next sections describe the AEDC Impulse Facility, the operation of the time-of-flight mass spectrometer, and progress made toward the design of the precision isokinetic sampling system the will be placed into the free stream of the test section.

AEDC Impulse Facility

The AEDC free piston shock tunnel (Impulse Facility), described by Maus (Ref. 6) and Blanks and DeWitt (Ref. 7), is a free-piston shock tunnel facility that can produce very high velocity, high stagnation pressure flows of very short duration. The free piston tunnel concept was developed by Stalker in Australia (Ref 2). The AEDC facility burns gun powder to generate combustion gases which are used to drive an expendable piston, which in turn adiabatically heats a light driver gas (helium). Helium is placed in the compression tube, downstream of the piston. The test gas, air, is placed in the shock tube, downstream of the diaphragm. Gun powder, located at the upstream end of the facility, is burnt and the combustion gases drive the heavy piston. The piston adiabatically compresses and heats the driver gas. At a given pressure the primary diaphragm bursts, initiating a shock wave that propagates down the shock tube and compresses the air. The shock wave reflects from the end of the shock tube, and propagates upstream, thereby compressing the air even further to stagnation conditions. The compressed air then ruptures the nozzle diaphragm and flow is initiated through the nozzle. Air, at high stagnation enthalpy and pressure, is then accelerated through the nozzle, yielding a steady flow in the test section. The test is terminated by the arrival of the driver gas (helium) in the test section, a few milliseconds after the initiation if the test section flow.

Typical test section conditions for the AEDC Impulse Facility are (Ref. 6, 8): Mach number = 7.3; velocity = 4.6 km/s; pressure = 2620 Pa; temperature = 1000 K; density = 0.009 kg/m³, dynamic pressure = 9.3×10^4 Pa. Major species in the flow stream: O₂, N₂, O and NO, the latter two being of lesser abundance.

Free Stream Gas Sampling System

A beam of free stream molecules must be isokinetically sampled and directed into the time-of-flight mass spectrometer to determine when helium reaches the test section. The beam will be formed from the molecules which pass through a series of precision isokinetic skimmers that are aligned with the test section stream (Ref. 5, 9). Isokinetic sampling is essential so that the sample represents the gas present at the skimmer tip, in the absence of the skimmer. Diffusive separation due to the density, pressure, and temperature gradients caused by the presence of the skimmer and other interfering effects must be examined. The gas sampling scheme used by Skinner (Ref. 5) is shown in Figure 1. The Skinner system has been used as a baseline for the present design. The flow through the first skimmer must be maintained in continuum flow (Ref. 10), which requires that the Knudsen number of the skimmer opening, based on the skimmer orifice diameter and the mean free path of the approaching molecules at stagnation conditions, must be maintained less than 0.05 or so. The first skimmer must swallow the shock and accelerate the flow. Flow through the opening on the third skimmer must be in free-molecule flow, which is defined as having a Knudsen greater than 2 (Ref. 11). This Knudsen number is based on the orifice diameter and the mean free path determined by the stagnation conditions of the flow that proceeded through the second skimmer. The molecular beam that is formed by the skimmer system will then be introduced into the mass spectrometer, which will be aligned perpendicular to the beam direction (Figure 3). The neutral molecules will be ionized and the ions will be detected by a microchannel plate (MCP) detector.

The system described above has been successfully used by

Skinner and Stalker (Ref. 9) in a free-piston shock tunnel to detect the driver gas (85% helium and 15% argon). They pulsed the TOF/MS electron beam at 18 kHz, yielding a full mass spectra each 55 microsec. Their flight tube was 1 m long and they detected the ions with an electron multiplier detector. The flow through the second and third skimmers was pumped by separate diffusion pumps.

The first skimmer must be designed to withstand the dynamic impulse generated by the formation of the shock wave at the initiation of the flow in the test section. Its included angle should be in the range of 30 to 60 degrees. Since the mass flow through the first skimmer is very great, it cannot be connected to a vacuum pump but instead its flow must be pumped by the external test section. The pressure downstream of the first skimmer must always be low enough for the shock wave to be swallowed while the pressure between the second and third skimmers must be low enough for the third skimmer to be in free-molecule flow. The third skimmer should be designed with an included angle of around 30 degrees with a thin wall to avoid collisions with the internal surface from interfering with the molecular beam that is formed (Ref. 11). Flow through the third and defining skimmer must be pumped by a large vacuum pump to maintain a background pressure in the mass spectrometer less than 10^{-6} torr. The pumping requirements and the desire to minimize the interference of the skimmer system with the test section flow place challenging constraints upon the design.

Time-of-Flight Mass Spectrometer

A Comstock Model TOF-101 time-of-flight mass spectrometer was purchased for measuring the time of arrival of the helium driver gas in the AEDC Impulse Facility. A time-of-flight mass spectrometer operates as follows. A beam of neutral molecules, formed by the isokinetic skimmer system, is drawn into an equipotential region at ground potential where it is struck by a pulsed beam of electrons of appropriate energy. The collisions with the electrons results in the partial ionization of the

molecules in the beam, forming pulses of positive ions. The positive ions are extracted by a high negative potential created at right angles to both the molecular beam and the electron beam. After extraction the ions drift in an equipotential flight tube that is maintained at a high negative potential. The ions strike a microchannel plate detector at the end of the flight tube, generating secondary electrons which are amplified to a level which can produce sufficient voltage drop across a resistor to be measured on an oscilloscope. Each mass acquires its own drift speed and arrives at the microchannel plate detector a distinctive time after the initiation of the pulse, yielding a voltage signal at a time that is characteristic of the molecular species.

The Comstock time-of-flight mass spectrometer consists of a pulsed electron-impact source, a flight tube, a microchannel plate detector, and associated power supplies and vacuum chamber. The flight tube is 1 meter long. The mass spectrometer can measure the presence of molecules over the mass range of less than 1 to 50,000 amu with a nominal resolution of 200 at $M/Z = 131$, where M is the molecular weight and Z the number of electrons removed from the molecule.

Electrons are generated by a filament and drawn by a pulsed anode into an equipotential region at which is at ground potential. The electron current is measured by a Faraday cup. The pulsed electron beam produces a repetitive pulse of ions. The ions are formed by collisions of the neutral molecules in the beam with the electrons. The ions are extracted and sent down the flight tube by a voltage pulse, applied to a repeller plate, that is initiated by a variable time delay after the initiation of the electron pulse. The width and repetition rate of both pulses can be independently varied, as well as the delay between them. The ions are accelerated by the entrance grid to the flight tube. Once in the equipotential flight tube they drift with a constant velocity before striking the microchannel plate detector..

Operation of the TOF Mass Spectrometer

All ions with the same charge acquire identical kinetic energy from the electric field. Then, assuming that they originally had zero kinetic energy, the time-of-flight signal is related to the mass of the molecule by the following equation

$$T - T_c = K\sqrt{M}$$

where T = the measured time-of-flight, sec
 T_c = correction to time-of-flight, sec
 M = molecular weight measured in amu

and K is a constant determined from the measurement of the time-of-flight of at least two known masses. Generally it is much better to use the time-of-flight from several known molecules in a regression analysis to determine the best value for the constants. The measured times for hydrogen, helium, water vapor, nitrogen, and argon produced the following regression equation with M given by in amu units:

$$t(\mu\text{sec}) = 1.514470\sqrt{M} + 0.300130$$

This equation was obtained from a gas input mixture of 85% nitrogen, and 5% each of hydrogen, helium, and argon. An amount of gas necessary to raise the background pressure by about 5×10^{-7} torr was introduced into the vacuum chamber. An accurate theoretical estimate of the flight time using dimensions provided by Comstock indicated that the slope in this equation should be 1.5458, an agreement to within 2% of the measured value.

The measurement of an unknown mass (M_u) is determined by comparing the measured time of an unknown mass with that for a known mass (M_k) through the relation

$$M_u = [M_k + \frac{(T_u - T_k)}{K}]^{\frac{1}{2}}$$

The molecular beam to be sampled from the AEDC Impulse Facility will have a mean energy of 3.15 eV in the beam direction, which is perpendicular to the time-of-flight direction. This

energy would cause a deflection of 3.5 cm perpendicular to the line-of-flight direction in the flight tube, and would cause the ions to miss the microchannel plate detector. X and Y deflection plates are provided to correct for this effect; their potentials must be adjusted until the ions strike the detector face.

An ion lens can be used to either focus or defocus the ion beam onto the microchannel plate detector. The voltage applied to the ion lens can be varied until the observed signal is optimized.

The ionization cross section due to electron collisions for a particular molecule rises from zero before the ionization potential energy to a maximum at an electron energy that depends upon the molecule and then slowly decreases. The ionization cross section for molecules of interest peak in the range of electron energies of 60 to 80 eV. For nitrogen the peak value is approximately equal to the neutral scattering cross section but the peak is smaller for helium by a factor greater than eight. This makes the detection of helium more difficult than other atmospheric gases.

The fragmentation pattern of the mass spectrometer depends upon the instrument. However, experience from other instruments indicates that, for a mixture of air and helium, one would expect to observe He, H₂O, OH, N₂, N, O₂, O, and minor amounts of other molecules.

An example of the mass spectra measured with the Comstock mass spectrometer in a bench application is shown in Figure 2. Helium was input to the vacuum chamber surrounding the mass spectrometer until the pressure was 7×10^{-7} torr. All of the other molecules shown in the spectra occurred in the background gas within the vacuum chamber. This spectra was obtained using settings which optimized the measurement of helium.

NPARC Computations

The general requirements that must be placed upon the skimmer system were discussed previously. It was pointed out that the successful operation of the skimmers depends upon having a detailed knowledge of the flow field about the first skimmer and at the

entrance to each skimmer orifice. Continuum (CFD) and molecular (Monte Carlo) computational techniques were used to estimate these flow characteristics. The skimmer system of Skinner and Stalker (Ref. 9), applied to the conditions of Run 21 in the AEDC Impulse Facility (Ref.8), was used as a baseline case. The continuum computations are described in this section and the Monte Carlo in the next.

The NPARC2D CFD code was used to compute the flow field approaching the first two skimmers. The skimmers had an external and internal half-angle of 30 degrees, with an internal chamfer of 20 degrees. The first skimmer was assumed to have a 2 mm diameter opening, the second a 0.7 mm diameter opening, and the separation between the two skimmers was 18.5 mm. The flow between the skimmers was allowed to freely communicate with the external test section flow, although in practice there will have to be a support structure between the first and second skimmers that will partially restrict this flow (see Figure 1).

The objective of the computations was to determine the conditions at the entrance to and external to the outer edge of the first skimmer. These conditions were then used as realistic upstream conditions for Monte Carlo computations for the flow through the following two skimmers. Monte Carlo computations were required for the flow through the latter two skimmers because the Knudsen number of the flow approaching these skimmers was expected to be large and their flows were not expected to be continuum. Most of the flow through the second skimmer will be removed by a large vacuum pump in the Impulse Facility application. This feature was not assumed for the continuum computations.

Conditions in the free stream used by the computations (Run 21 from the AEDC Impulse Facility, Ref. 8) were as follows: gas - air; Mach number - 7.295; free stream velocity - 4630 m/s; free stream pressure - 2620 Pa; free stream temperature - 999.2 K; free stream density - 0.008804 kg/m³; free stream gamma - 1.355; mixture gas constant - 297.60 J/kg K; Reynolds number per mm - 136.3.

The multi-block version of NPARC2D was used to compute the flow field. A seven block grid was generated using the grid generation program GRIDGEN. The free stream conditions were used for the reference conditions for the NPARC2D code. Only inviscid computations were performed.

Computed pressure contours in the vicinity of the first two skimmers are shown in Figure 3. Since the approaching Mach number is very large, the shock wave is close to the first conical skimmer and attached to the lip of the orifice. Because of the expansion of the flow around the lip and the pumping action of the region between the first two skimmers, no shock appears ahead of the skimmer orifice. Thus, the shock wave has been swallowed. Expansion around the upper surface of the first skimmer assists in pumping the flow through the first skimmer and swallowing the shock wave. While Skinner predicted a recovery shock between the first two skimmers (Ref. 5), the present computations did not predict a shock between the skimmers.

The computations indicate that the inlet conditions are very close to the conditions in the approaching stream over about 80% of the orifice of the first skimmer. This should be adequate for isokinetic sampling through the following skimmers, although the diffusive separation caused by the strong gradients in the region of the skimmer lip must be examined further. The computed velocity remains very close to the free-stream value across the face of the second skimmer, but the density diminished greatly and fell off from the center line value. The computations across the second skimmer orifice are not considered to be accurate because of the low density and will be repeated using the Monte Carlo code. Integration of the computations predicted that the mass flow through the first orifice is 0.00012 kg/s, which translates to 2.6×10^{21} molecules/s. Likewise, the predicted mass flow through the second skimmer is 8.5×10^{-6} kgm/s, or 1.8×10^{20} molecules/s.

An integration of the surface pressure on the first skimmer predicts that the steady free stream applies a force of 94.3 N (21.2 lb) to the outer skimmer surface. The skimmer design also

requires an estimate of the impulse applied to the skimmer during the initiation of the flow in the test section.

The NPARC2D computations indicate that 7% of the mass flow through the first skimmer also proceeds through the second skimmer. This prediction, which is important for sizing the vacuum pumps, will later be compared to the prediction from the Monte Carlo computations.

Monte Carlo Computation

Monte Carlo computations were performed using the code DSMC2D-MP developed by Dr. Tim J. Bartel at Sandia National Laboratory. This is a parallel version of DSMC and was run on the NCUBE machine at Sandia.

The Monte Carlo computations required that some initial files be prepared. These included an inlet file, a species file, and an input file. The inlet file contains the inflow conditions which were obtained from the NPARC2D solution for density, temperature, and velocity across the orifice face and over the top of the first skimmer. The inlet file also required the vibrational and rotational temperatures and composition along the inlet flow surface. The NPARC2D solution had been obtained assuming that the incoming stream was equilibrium air. To model the flow in the Impulse Facility for the initial arrival of the helium driver gas, the composition was modified to include 1% helium (mass fraction) mixed with the air. The appropriate mixture properties were obtained from the One Dimensional Equilibrium (ODE) code using $T = 1000. \text{ K}$, $p = 2620 \text{ Pa}$. The ODE computed results were as follows: free stream density - 0.00857 kg/m^3 ; free stream sound speed - 641.5 m/s ; gamma - 1.346 ; mole fractions - $\text{He} = 0.0656$, $\text{NO} = 0.05551$, $\text{O}_2 = 0.1283$; $\text{N}_2 = 0.682$; $\text{O} = 0.06859$. The number density is related to mass density by the relation

$$n = 2.296 \times 10^{25} \rho$$

which is obtained using the mixture molecular weight from ODE of 26.24. The rotational and vibrational temperatures were assumed to

be equal to the translational temperature in the initial profile.

The species file requires information about the molecular diameters and the internal energy states and energy transfer efficiencies during collisions. The molecular diameters were supplied by J. E. Johannes of Sandia National Laboratory. They were obtained from the transport properties of the various molecules at 1000 K given in the Lawrence Livermore code CHEMKIN.

The geometry and other conditions are specified in the input file. The geometry must be broken into a number of four-sided regions. These regions are defined by global corner points, whose location is specified in a table in the input file. The input file contains information about each region, including boundary condition specification and the number of computational cell in each region. The present problem was broken into fifteen regions with 16,450 computational cells. The code contains provision to connect only one region to a vacuum pump but in the present problem three regions will be connected to vacuum pumps. Entries in the inlet file had to be changed until the number of numerical molecules was adequate for the problem (on the order of 300,000).

The code includes an initialization code that generates the computational grid and the initial start file. It also gives an indication of the number of computational molecules. If that number is not adequate the input file must be edited until an adequate number are generated. Also, the grid should be examined to make sure it has been specified correctly before the Monte Carlo computations are initiated.

The dsmc2d code was initially run on a Silicone Graphics work station. 100,000 time steps took approximately two days to complete. Later computations were performed on the NCUBE machine at Sandia National Laboratory. After running for the specified number of time steps, the program generates output files. A post-processing utility is used to examine the computational results given in the output files. Post-processing will generate a file that can be examined using the plotting program TECPLOT. If the computations are proceeding in a satisfactory manner, then they may

be continued using the restart code. The restart code will generate a file that can be used to restart the dsmc code. The restart code computes a Maxwellian distribution in each computational cell rather than saving the position and velocity of each particle from the previous computation. Thus it is advisable to run a number of unsteady time steps at the beginning of the restart to get back to the original nonequilibrium state.

Contours of axial velocity computed after 400,000 time steps are shown in Figure 4 and pressure contours in Figure 5. The shock formed on the front surface of the second skimmer is clearly visible. The flow then expands through the second skimmer and only the faintest hints of a shock appear upstream of the final 30 degree included angle skimmer (0.7 mm diameter orifice). However, the beam formed by the last skimmer is very noticeable. The lack of beam structure in the downstream region which is connected to the vacuum pump is an indication of poor statistics due to too few particles passing through the second skimmer. That number will be increased for the next round of computations. The pressure varies from 4600 Pa to 3×10^{-5} Pa and this variation is difficult to show in the figure. The results shown are too preliminary to base a skimmer design upon; the computations are not converged and will be continued.

Concluding Remarks

A program has been initiated for the design of a system that will measure the time-of-arrival of the driver gas in the AEDC Impulse Facility. A number of things remain to be done before the skimmer system can be designed and installed in the facility. An effusive source calibration system for the TOF/MS must be constructed and used to obtain fragmentation patterns that apply to the Comstock instrument. More skimmer geometries must be examined computationally to determine the best geometry to minimize the interference effects on the measurement, and to determine geometries that can be constructed with accuracy and that will provide the required conductance into the vacuum pumps. The Monte

Carlo code must be modified to allow multiple regions to be connected to independent vacuum pumps. Finally, the skimmer system must be designed and constructed. It will be advisable to test the final design in a shock tube that can be operated at a minimal cost before the system is installed in the Impulse Facility. These tasks will be performed in the next phase of this work.

Acknowledgements

The author wishes to express his appreciation to Sverdrup, Inc. employees at AEDC who contributed significantly to this project. The project was under the supervision of Mr. Paul Jalbert. The author especially appreciates the assistance of Mr. Robert Hiers III and Mr. Bob Hiers, Jr. NPARC computations were assisted by Dr. Greg Powers. The supervision of Dr. Robert Reed was essential for the work. DSMC computational assistance was provided by Dr. Timothy J. Bartel of Sandia National Laboratory.

References

1. Stalker, R. J., "Hypervelocity aerodynamics with chemical nonequilibrium," Ann. Rev. Fluid Mech. Vol. 21, pp. 37-60, 1989.
2. Stalker, R. J., "The free-piston shock tube," The Aero. Quart., Vol. XVII, pp. 351-370, 1966.
3. Simmons, J. M., "Measurement techniques in high-enthalpy hypersonic facilities," Exp. Thermal and Fluid Science, Vol. 10, pp. 454-469, 1995.
4. Crane, K. C. A. and Stalker, R. J., "Mass-spectrometric analysis of hypersonic flows," J. Phys. D: Appl. Phys., Vol 10, pp. 679-695, 1977.
5. Skinner, K. A., "Species measurements using mass spectrometry in high Mach number flows," 11th Australasian Fluid Mechanics Conference, Univ. of Tasmania, Hobart, Australia, 14-18 December, 1992.
6. Maus, J. R., "A mathematical model for the g-range impulse facility," AEDC-TMR-91-V10, October, 1991.
7. Blanks, J. R. and DeWitt, J. R., "Calibration tests of

AEDC free-piston shock tunnel," AIAA-94-2526, presented at 18th AIAA Aerospace Ground Testing Conference, June 20-23, 1994.

8. Maus, J. R., AEDC, private communication.

9. Skinner, K. A. and Stalker, R. J., "Time-of-flight mass spectrometry for impulse facilities," AIAA J., Vol. 32, No. 11, pp. 2325-2328, 1994.

10. Skinner, G. T., "Molecular beam for the study of high-temperature-gas collision processes," Phys. Fluids, Vol. 4, No. 9, 1961, pp. 1172-1176.

11. Bird, G. A., "Transition regime behavior of supersonic beam skimmers," Phys. Fluids, Vol. 19, No. 10, 1976, pp. 1486-1491.

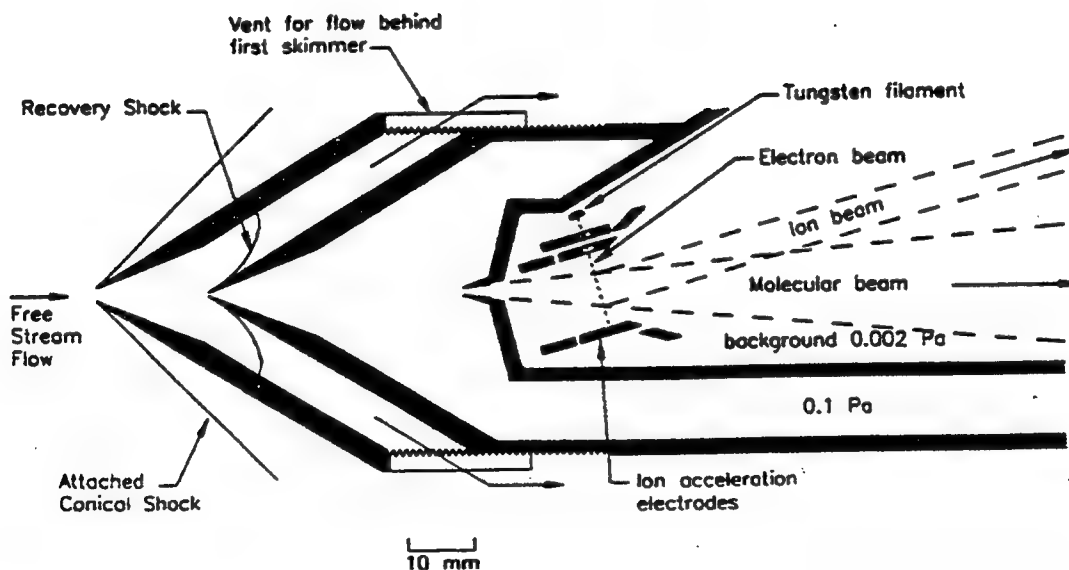


Figure 1. Skimmer arrangement and ionizer used by Skinner (Ref. 5 and 9).

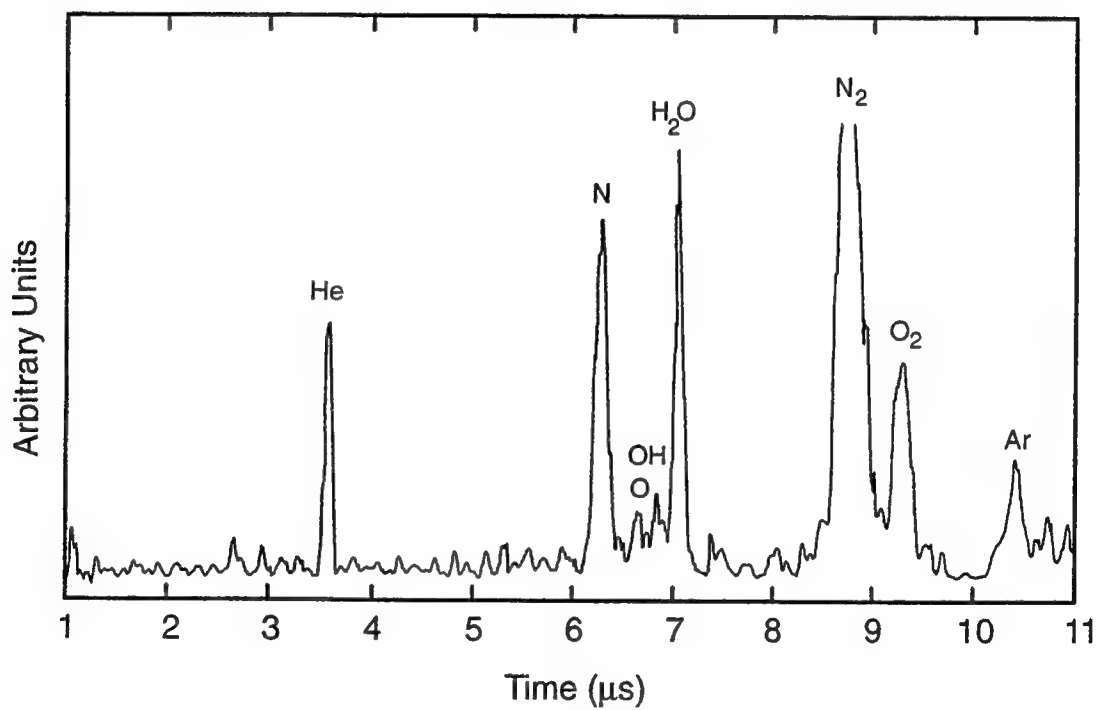


Figure 2. Spectra 8, taken June 30, 1995, from an input of pure helium with the mass spectrometer optimized for measuring helium.

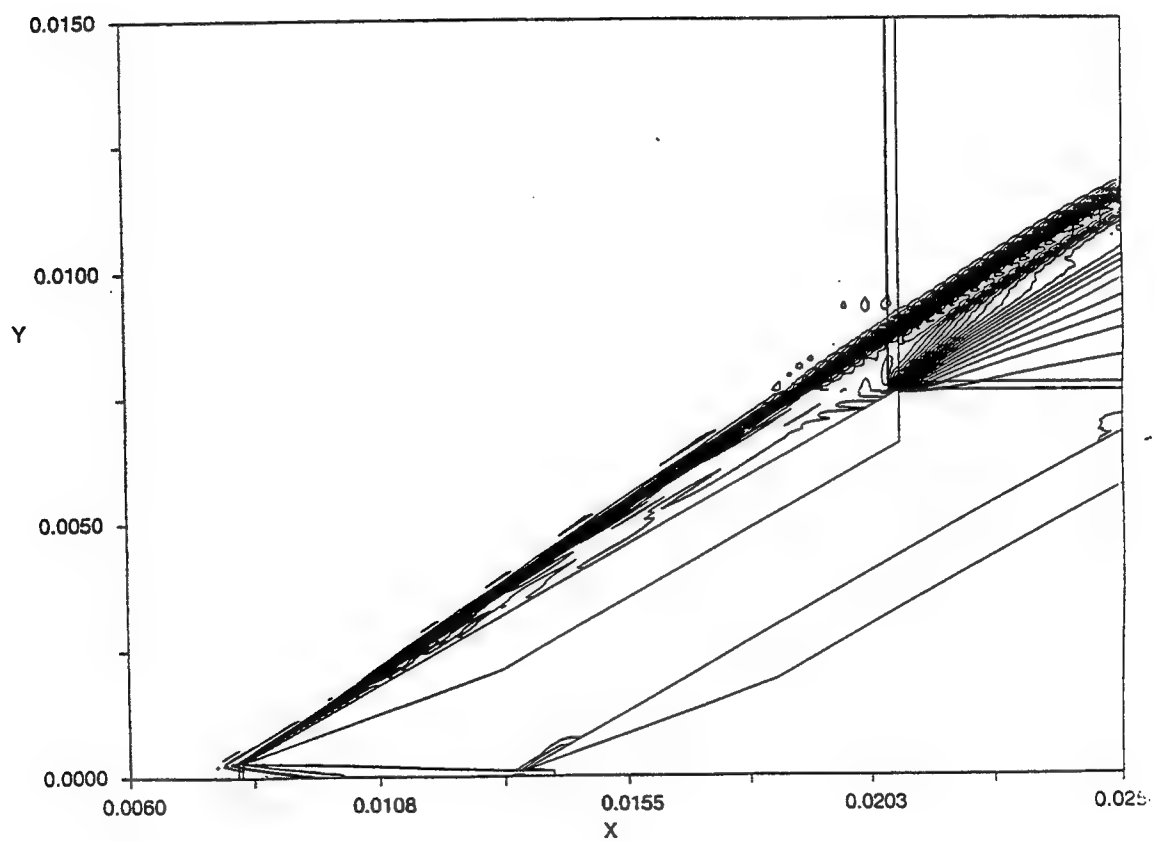


Figure 3. Pressure contours computed by NPARC2D; close-up of the flow over the first skimmer.

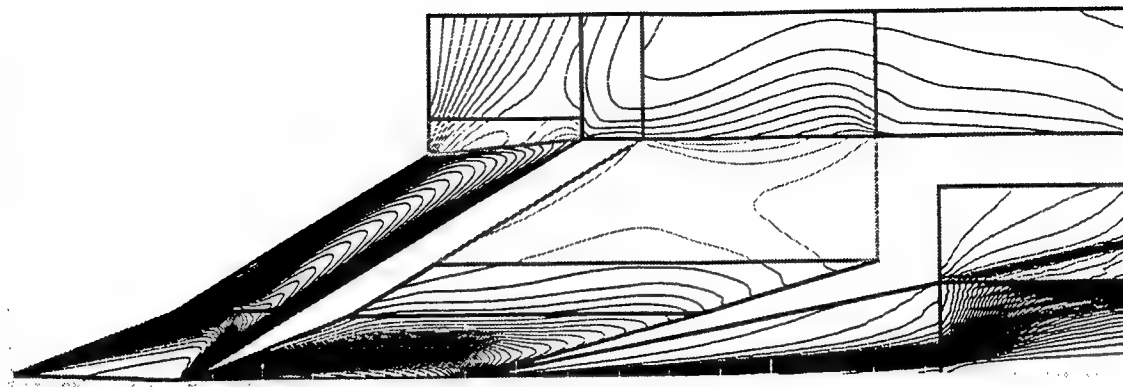


Figure 4. Axial velocity contours computed by the Monte Carlo code.

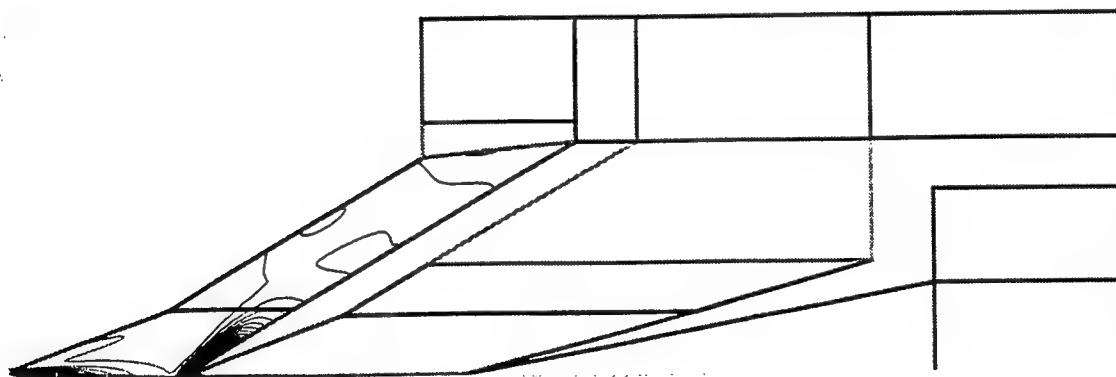


Figure 5. Pressure contours computed by the Monte Carlo code.

**MULTIRESOLUTION SCHEMES
FOR SOLVING
CONSERVATION LAWS**

**Peter R. Massopust
Visiting Assistant Professor
Department of Mathematics**

**Sam Houston State University
Huntsville, TX 77341**

**Final Report for:
Summer Faculty Research Program
Arnold Engineering Development Center**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC**

and

Arnold Engineering Development Center

August 1995

MULTIRESOLUTION SCHEMES
FOR SOLVING
CONSERVATION LAWS

Peter R. Massopust
Visiting Assistant Professor
Department of Mathematics
Sam Houston State University

Abstract

Multiresolution schemes for the numerical solution of nonlinear partial differential equations of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot f(u) = 0,$$

where u and f are vector-valued functions were investigated. These schemes are based on resolving the numerical fluxes corresponding to f on a nested sequence of increasingly coarser grids and reconstructing them using a combination of direct evaluation and interpolation. This procedure decreases the number of direct (expensive) flux evaluations and this results in a computational speed-up.

MULTIRESOLUTION SCHEMES FOR SOLVING CONSERVATION LAWS

Peter R. Massopust

1 Introduction

Multiresolution schemes were first introduced by the late A. Harten in a series of papers. (An incomplete list is [3, 4, 5], and [6].) Since these schemes are based on decomposing and reconstructing functions employing a compression technique, they are closely related to orthogonal and biorthogonal wavelets ([1, 2]). To a certain degree these schemes extend the theory of wavelets.

Multiresolution analyses based on orthogonal or biorthogonal wavelets have been very successful in resolving the local fine structure of functions which exhibit varying degrees of smoothness across their domains. The main ingredients for a multiresolution analysis are: a nested sequence of approximation spaces on which a given function is to be resolved, and a decomposition and reconstruction algorithm. We refer to [1] or [2] for more details.

Instead of starting with a nested sequence of approximation spaces $\{V_k\}_{k \in \mathbb{Z}}$ for a function f in a given function space \mathcal{F} and constructing projection operators $\mathcal{P}_k : \mathcal{F} \rightarrow V_k$, $k \in \mathbb{Z}$, which ultimately yield the representation of f in terms of wavelet coefficients, Harten began with a function space \mathcal{F} and a sequence of *nested discretization operators* $\{\mathcal{D}_k\}_{k \in \mathbb{N}_0}$; each \mathcal{D}_k assigns a sequence of numerical values to a function $f \in \mathcal{F}$. Decomposition and reconstruction algorithms are then defined directly on the *discretization spaces* $V_k := \mathcal{D}_k(\mathcal{F})$, $k \in \mathbb{N}_0$.

Such a sequence of discretization operators is, for instance, obtained as follows.

Example 1 Let $\{\Gamma_k\}_{0 \leq k \leq L}$, $L \in \mathbb{N}$, be a given sequence of grids each contained in some bounded region $\Omega \in \mathbb{R}^n$ with the property

$$\Gamma_0 \prec \Gamma_1 \prec \cdots \prec \Gamma_L,$$

where \prec denotes an ordering relation on the grids: $\Gamma_k \prec \Gamma_{k+1}$ iff $x \in \Gamma_{k+1} \implies x \in \Gamma_k$. In other words, grid Γ_k is finer than grid Γ_{k+1} . Now let f be a function whose domain includes Ω . For $k = 0, \dots, L$, define \mathcal{D}_k by

$$\mathcal{D}_k(f) := \{f(x_j^k) : x_j^k \in \Gamma_k\}. \quad (1)$$

(In the above equation it is assumed $x_0^k < x_1^k < \cdots < x_{J_k}^k$, where the cardinality of Γ_k is $1 + J_k$.) In other words, $\mathcal{D}_k(f)$ is the vector whose components are the evaluations of the function f at the grid point x_j^k .

This type of discretization operator appears, in implicit form, in the numerical solution techniques for conservation laws of the form $\partial u / \partial t + \nabla \cdot f(u) = 0$. The idea is to employ the decomposition and reconstruction algorithms obtained from discretization operators to compress the numerical solution and the numerical fluxes in an efficient way to reduce CPU time but without losing the accurate description of the local fine structure of the exact solution. In regions where the function exhibits a high degree of regularity only a relatively small number of numerical values is needed to describe it, whereas in regions where the function exhibits discontinuities or steep gradients a larger number is required.

The outline of this report is as follows. In the next section the multiresolution schemes are introduced and the reconstruction and decomposition algorithms are defined. In Section 3 it is shown how such multiresolution schemes may be used to obtain numerical solution to conservation laws. In this section our results and ongoing work is referenced. The last section provides a summary and addresses some open problems and questions.

2 Multiresolution Schemes

In this section Harten's multiresolution schemes are introduced and the main ideas are illustrated by considering the type of discretization operators introduced in Example 1. The presentation of these schemes is not the most general one, but for the purposes of this report it is sufficient. For a more detailed and general introduction we refer the interested reader to [4].

Let \mathcal{F} be a Banach space of functions with (denumerable) basis. For the applications that are mentioned in this report \mathcal{F} may be chosen to be the linear space $C(\Omega)$ of continuous functions on a bounded region $\Omega \subset \mathbb{R}^n$ with the sup-norm $\|f\|_\infty := \sup \{|f(x)| : x \in \Omega\}$, or the Sobolev space $H^s(\Omega)$, $s < n/2$, with norm $\|f\|_{H^s} = (\sum_{|\nu| \leq s} \int_\Omega |D^\nu f|^2 dx)^{1/2}$. Here fractional derivatives are understood in the Riesz sense.

Definition 1 A collection $\{\mathcal{D}_k\}$, $k = 0, 1, \dots, L$, $L \in \mathbb{N}$, of linear operators on \mathcal{F} is called a nested sequence of discretization operators (on \mathcal{F}) if

$$\mathcal{D}_k(f) = 0 \implies \mathcal{D}_{k+1}(f) = 0, \quad \text{for all } k = 0, 1, \dots, L, \quad (2)$$

and the range of \mathcal{D}_k is a linear space with denumerable basis.

In this report the concentration is on the discretization operators introduced in Eqn. (1). However, besides these *point-value operators* there are also cell-averaging operators and wavelet-based operators (see, for instance, [4]).

It is easy to see that the range of the discretization operator \mathcal{D}_k defined by Eqn. (1) is \mathbb{R}^{J_k+1} , hence finite-dimensional. For each operator \mathcal{D}_k there exists a reconstruction operator \mathcal{R}_k which maps the range of \mathcal{D}_k into \mathcal{F} .

Definition 2 Let V be the range of the discretization operator \mathcal{D} . The right inverse \mathcal{R} of the discretization operator \mathcal{D} is called a reconstruction operator. I.e., $\mathcal{R} : V \rightarrow \mathcal{F}$ and

$$\mathcal{D}\mathcal{R}(v) = v, \quad \text{for all } v \in V. \quad (3)$$

It is worthwhile remarking that \mathcal{R} need not be a linear operator. Moreover, \mathcal{R} maps into \mathcal{F} , hence the range of \mathcal{R} is a subset of \mathcal{F} . (In case \mathcal{R} is linear, $\mathcal{R}(V)$ is a linear subspace of \mathcal{F} .)

Throughout this report reconstruction operators are assumed to be linear.

Example 2 For the sequence of discretization operators \mathcal{D}_k in Example 1 it is easy to obtain an associated sequence of reconstruction operators \mathcal{R}_k : Given any sequence of values $\{f(x_j^k)\}$ on grid Γ_k , a right inverse of \mathcal{D}_k is an operator that assigns to these values any function f^* that interpolates on the set $\{(x_j^k, f(x_j^k)) : x_j^k \in \Gamma_k\}$. This clearly shows that \mathcal{R}_k is not uniquely determined and that its range is only a subspace of \mathcal{F} .

There are two other operators which are related to \mathcal{D}_k and \mathcal{R}_k .

Definition 3 The linear operator $D_{k-1}^k : V_{k-1} \rightarrow V_k$ defined by

$$D_{k-1}^k := \mathcal{D}_{k-1}\mathcal{R}_k, \quad \text{for all } k \quad (4)$$

is called a decimation operator.

The linear operator $P_k^{k-1} : V_k \rightarrow V_{k-1}$ given by

$$P_k^{k-1} := \mathcal{D}_k\mathcal{R}_{k-1}, \quad \text{for all } k \quad (5)$$

is said to be a prediction operator.

Example 3 Using the discretization and reconstruction operators defined in Examples 1 and 2 above, the decimation and prediction operators are given by

$$D_{k-1}^k(f(x_j^{k-1})) = f(x_{2j}^{k-1}), \quad (6)$$

and

$$P_k^{k-1}(f(x_j^k)) = \begin{cases} f(x_{2j}^{k-1}) \\ \text{arbitrary} \end{cases}, \quad (7)$$

respectively.

It is not hard to see that the operator $\mathcal{R}_k\mathcal{D}_k$ maps the space \mathcal{F} onto a subspace of itself and, thus, the function $\mathcal{R}_k\mathcal{D}_k(f)$ is an approximation to $f \in \mathcal{F}$ (in the norm of \mathcal{F}). From an interpolation-theoretic point of view, the sequence of approximations $\{\mathcal{R}_k\mathcal{D}_k(f)\}_k$ should converge to f in the norm of \mathcal{F} . This

can be assured by imposing certain stability conditions on this sequence. (The interested reader is referred to [4] for complete details.) This obviously restricts the choice of reconstruction/prediction operators. In what follows we describe the relevant issues for the discretization operators defined in Example 1 and refer to the references for the general case.

It is clear that the arbitrariness in choosing values of f at oddly-indexed grid points needs to be removed (see Eqn. (7)). This can be done by interpolation: Let $\mathcal{I}_{k-1}(\cdot)$ denote any interpolation operator at those grid points and set

$$f(x_{2j}^{k-1}) = f(x_j^k), \quad \text{and} \quad f(x_{2j-1}^{k-1}) = \mathcal{I}_{k-1}(f(x_j^k)). \quad (8)$$

For this report a linear interpolation operator is chosen (nonlinear interpolation operators yield nonlinear reconstruction operators!):

$$\mathcal{I}_{k-1}(f(x_j^k)) := \frac{1}{2}(f(x_{j-1}^{k-1}) + f(x_j^k)). \quad (9)$$

The error between the exact value of f at x_{2j-1}^{k-1} and the interpolated value is measured by

$$d_j^k := f(x_{2j-1}^{k-1}) - \mathcal{I}_{k-1}(f(x_j^k)). \quad (10)$$

To ease notation, the following quantities are introduced:

$$v_j^k := f(x_j^k), \quad j = 0, 1, \dots, J_k; \quad k = 0, 1, \dots, L. \quad (11)$$

It is straight-forward to verify that the vector v^{k-1} uniquely determines the vectors v^k and d^k , and vice-versa. The decomposition can be carried out from the finest to the coarsest grid Γ_L ; this procedure then yields the *multiresolution decomposition* of v^0 and, consequently, f :

$$\begin{aligned} f &\sim v^0 \\ &\sim (v^1, d^1) \\ &\vdots \\ &\sim (v^L, d^L, \dots, d^1) \end{aligned}$$

(The symbol \sim denotes the representation of f on discretized data.) Starting on the coarsest grid with the vectors v^L, d^L, \dots, d^1 , and moving up towards the finest grid produces the *multiresolution reconstruction* of v^L .

Instead of reconstructing v^0 exactly from v^L, d^L, \dots, d^1 , it is sometimes desirable to consider *lossy* reconstruction: Given a threshold $\epsilon > 0$, determined by numerical constraints or other criteria, an approximation f_ϵ to f is obtained employing the following algorithm: Given an $\epsilon > 0$, determine an ϵ_k , $k = 0, 1, \dots, L$, on each grid Γ_k such that $\sum \epsilon_k = \epsilon$. (A natural choice for the ϵ_k is $\epsilon_k = 2^{-k}\epsilon$.)

$$\text{If } |d_j^k| < \epsilon_k$$

$$\begin{aligned}
& v_{2j-1}^{k-1} = \mathcal{I}_{k-1}(v_j^k) \\
& \text{else} \\
& v_{2j-1}^{k-1} = v_{2^{k-1}(2j-1)}^0.
\end{aligned}$$

It can be shown that under the above criterion $\|f - f_\epsilon\| < \epsilon$ (in the norm of \mathcal{F}) (cf., for instance, [4]).

The above algorithm also provides a means of detecting discontinuities in f and its derivatives. Interpolation-theoretic arguments show that if f has a jump discontinuity in the s th derivative at \tilde{x} , then if x_j^k is in a neighborhood of \tilde{x}

$$d_j^k(v^0) \approx \begin{cases} (h_k)^s [f^{(s)}] & 0 \leq s \leq r \\ (h_k)^r f^{(r)} & s > r, \end{cases}$$

where $[\cdot]$ denotes the jump of at the discontinuity of $f^{(s)}$, r the order of approximation, and $h_k = \min\{x_j^k - x_{j-1}^k : j = 1, \dots, J_k\}$ (see also [3]). The above observation implies that if grid Γ_k is fine enough then

$$|d_{2j}^{k-1}| \approx 2^{-\sigma} |d_j^k|, \quad \sigma = \min s, r. \quad (12)$$

Therefore, in regions where the function f and its derivatives are continuous, a decrease in the mesh size of the grid results in a reduction of d_j^k , whereas in the neighborhood of a jump discontinuity or a steep gradient the modulus of d_j^k is independent of the grid spacing. As in wavelet theory, the reconstruction algorithm can thus be used to investigate the local regularity of f .

Pursuant to the above argument it is therefore necessary to determine all points x_j^k at which the error coefficient d_j^k is larger than ϵ . The procedure for performing this task is as follows (see also [3, 4]): Define a matrix $A = (a_{kj})_{\substack{1 \leq k \leq L \\ 1 \leq j \leq J_k}}$ whose entries consists of zero's and one's according to

$$\begin{aligned}
& \text{If } |d_j^k| \epsilon_k & a_{kj} &= 0 \\
& \text{else} & a_{k,j \pm \ell} &= 1 \\
& \text{if also } |d_j^k| \geq 2^{s+1} \epsilon_k & a_{k-1,2j \pm \ell} &= 1.
\end{aligned}$$

The quantity ℓ is determined by the CFL number. It serves to keep track of the direction along which the shock/jump discontinuity is moving and how fast this movement is. The matrix A may now be implemented into the (lossy) reconstruction algorithm to flag all grid points at which interpolation is not advisable. The sparseness of the matrix A is paramount to the number of times interpolation is used over exact evaluation on the finest grid Γ_0 . Since, from a computational viewpoint, interpolation is "cheaper" than exact evaluation a

highly sparse matrix A is an indicator for the possibility of computational speed-up. A. Harten in a series of numerical experiments (see the references in the bibliography) has shown that compressions ratios of 5 to 8 are quite feasible. (By compression ratio is meant the ratio of zero entries to one entries in A .)

3 Applications to Conservation Laws

In this section it is shown how the lossy reconstruction algorithm based upon the matrix A may be used to efficiently obtain a numerical approximation to the exact solution.

The type of conservation laws considered is of the form

$$\frac{\partial q}{\partial t} + \nabla \cdot f(q) = 0, \quad (13)$$

or, in discretized one-dimensional form on a fixed grid Γ_0 ,

$$q_j^{n+1} = q_j^n + \lambda (\bar{f}_j - \bar{f}_{j-1}). \quad (14)$$

Here q is in general a vector whose components may include the density ρ , the velocity u , and the pressure p . Furthermore, \bar{f} denotes the numerical flux, λ is the ratio of time to space mesh, and the subscript j refers to the j th grid point of Γ_0 . The arguments in the previous section suggest that the lossy reconstruction algorithm may be useful in calculating the expensive numerical fluxes in Eqn. (14). Adhering to Harten, a possible strategy for compressing the numerical solution q and the numerical fluxes \bar{f} is as follows:

- Determine the function space the solution is expected to belong to;
- Set up a sequence of progressively coarser grids $\{\Gamma_k\}$;
- Choose a sequence of discretization operators and an interpolation operator;
- Obtain the multiresolution representation of q and calculate the entries in the matrix A ;
- Evaluate the numerical fluxes exactly on the coarsest grid Γ_L :

$$\bar{f}_j^L = f(q_{2^L j}^0);$$

(The time dependence of q and \bar{f} is suppressed!)

- On grid Γ_k , $k = L - 1, \dots, 1$, the values of the fluxes at x_{2j}^{k-1} are equal to the values at x_j^k .
If $a_{kj} = 0$ interpolate the fluxes at x_{2j-1}^{k-1} from the values on the previous coarser grid; otherwise evaluate \bar{f} from the finest grid: $\bar{f}_{2j-1}^{k-1} = \bar{f}(q_{2^{k-1}(2j-1)}^0)$.

- Calculate q^0 at time $t = (n + 1)\Delta t$. This can be done by using an explicit or implicit time stepping method. The former was employed by Harten, in the report we used the latter.

The above strategy was employed to several conservation laws:

1. Riemann problem for Burgers equation:

$$q_t + (\frac{1}{2}q^2)_x = 0, \quad 0 \leq x \leq 1, \quad t > 0,$$

$$q(x, 0) = \begin{cases} 1 & 0 \leq x \leq 1/2 \\ 0 & 1/2 < x \leq 1; \end{cases}$$

2. Periodic Riemann Problem for Burgers equation:

$$q_t + (\frac{1}{2}q^2)_x = 0, \quad 0 \leq x \leq 1, \quad t > 0,$$

$$q(x, 0) = \begin{cases} 1 & 1/4 \leq x \leq 3/4 \\ 0 & \text{else}; \end{cases}$$

3. Periodic Initial Value Problem for Burgers equation:

$$q_t + (\frac{1}{2}q^2)_x = 0, \quad 0 \leq x \leq 1, \quad t > 0,$$

$$q(x, 0) = 2 + \sin(\pi(2x - 1));$$

4. Periodic Initial Value Problem for Burgers equation with a stiff source term ψ :

$$q_t + (\frac{1}{2}q^2)_x = 0, \quad 0 \leq x \leq 1, \quad t > 0,$$

$$q(x, 0) = 2 + \sin(\pi(2x - 1)),$$

$$\psi(x) = -\mu(x - 1)(x - 2)(x - 3);$$

5. Shock tube problem with $q = (\rho, \rho u, e)^T$ and $f(q) = (\rho u, \rho u^2, u(e + p))^T$ (T denotes the transpose).

For all problems an implicit first order scheme based on a Jacobi-type relaxation method was used. More precisely, in Problems 1- 4, the Jacobian in the residual was replaced by a linear approximation of the form

$$|\text{Jac}(R)| \approx \frac{1}{\Delta t} + \frac{|q^{n+1}|}{\Delta x} + |\psi'|, \quad (15)$$

where R is the residual $q_j^{n+1} - q_j^n - \lambda(\bar{f}_j - \bar{f}_{j-1}) - \psi_j'$, $j = 0, 1, \dots, J_0$. In case the source term ψ is absent, the last expression in the above equation is to be discarded. For Problem 5 a first-order implicit scheme of the above type was employed with \bar{f} replaced by

$$\tilde{f}_j = \frac{b_{\max}\bar{f}_j - b_{\min}\bar{f}_{j-1} + b_{\max}b_{\min}(q_{j+1} - q_j)}{b_{\max} - b_{\min}}, \quad (16)$$

where $b_{\min} = \min\{0, u_j - a_j\}$, $b_{\max} = \max\{0, u_{j+1} + a_{j+1}\}$, and $a_j = \sqrt{(\gamma p_j)/\rho_j}$, (γ polytropic gas constant), and

$$|\text{Jac}(R)| \approx \frac{1}{\Delta t} + \frac{|u_j^{n+1}| + a_j + c_j + c_{j-1}}{\Delta x}, \quad (17)$$

where $c_j = |(b_{\max} b_{\min})/(b_{\max} - b_{\min})|$.

The results of a lossy reconstruction are listed in the figures below. The top part shows the approximate numerical solution and the bottom part those grid points for which $a_{kj} = 1$. (Note that only grids Γ_L through Γ_1 are given.) Comparing the CPU times between the above lossy reconstruction and the straightforward evaluation of fluxes on a finest grid showed that in Problems 1 - 4 a computational speed-up of about 15% was achieved, whereas in Problem 5 the speed-up was about 20%. These results are for a *first order* implicit scheme and are expected to improve when higher order schemes are used.

4 Summary and Open Questions

The results can be summarized as follows:

- Lossy reconstruction algorithms for multiresolution schemes provide an efficient means to obtain accurate fine structure descriptions of solutions of conservation laws that exhibit shocks or steep gradients.
- A computational speed-up for first order accurate implicit schemes is achieved due to the sparseness of the matrix A . The trade-off between additional calculations (multiresolution decompositions, calculations of d_j^k , a_{kj} , etc.) and expensive flux calculations on a finest grid is more apparent in cases of complex numerical fluxes.
- A more significant computational speed-up is expected for higher order accurate schemes and problems involving a large number of chemical species, such as combustion problems.

The schemes investigated in this report provide a *basis* for further studies. The author is currently implementing higher order upwind schemes as, for instance, described in [10] to obtain lossy reconstruction algorithms for combustion problems including source terms.

Investigations of how to efficiently extend multiresolution schemes to conservation laws in two and three dimension, as well as a thorough study of the theoretical interpolation and approximation issues have also begun.

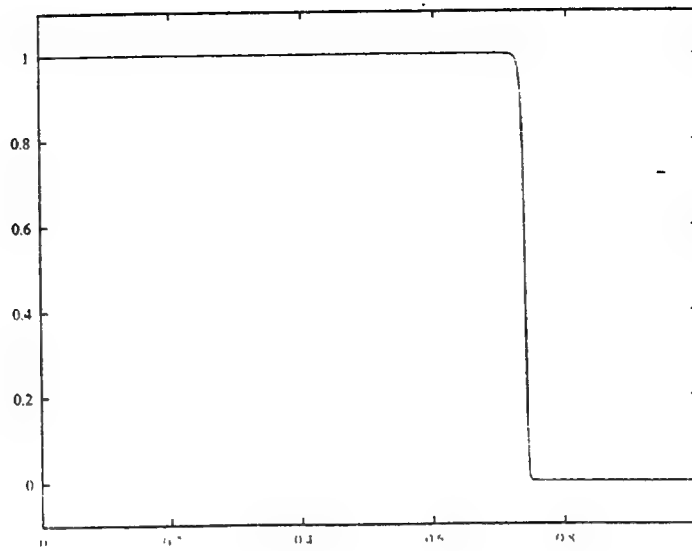


Figure 1: Problem 1 with $J_0 = 256$, $L = 5$, $\epsilon = 10^{-3}$, CFL = 0.8, and $n = 75$.

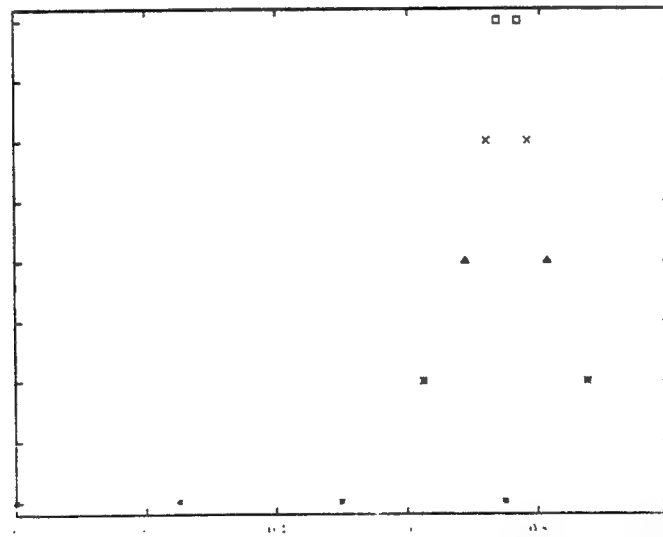


Figure 2: Problem 1: Flagged grid points

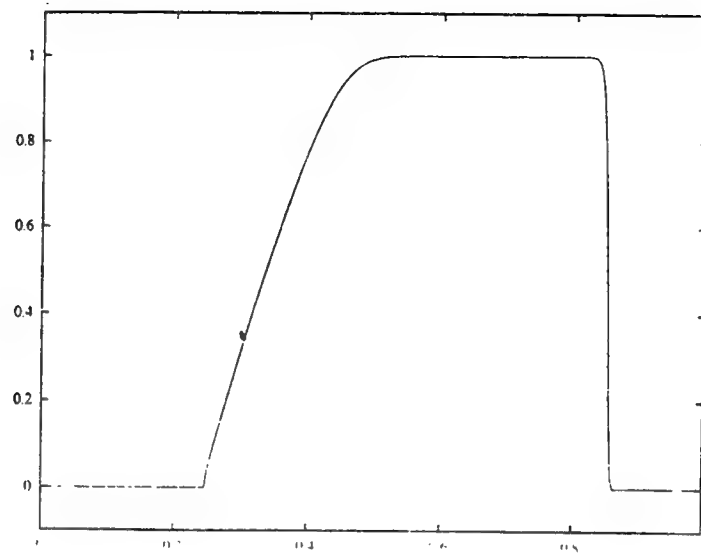


Figure 3: Problem 2 with $J_0 = 256$, $L = 5$, $\epsilon = 10^{-3}$, CFL = 0.8, and $n = 50$.

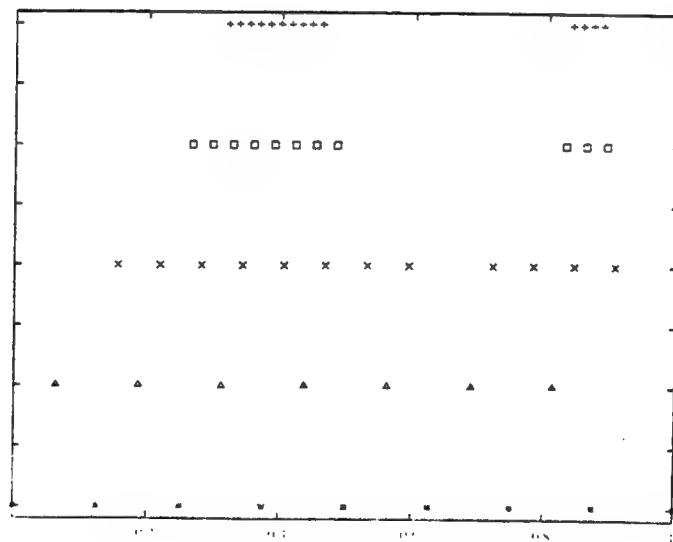


Figure 4: Problem 2: Flagged grid points

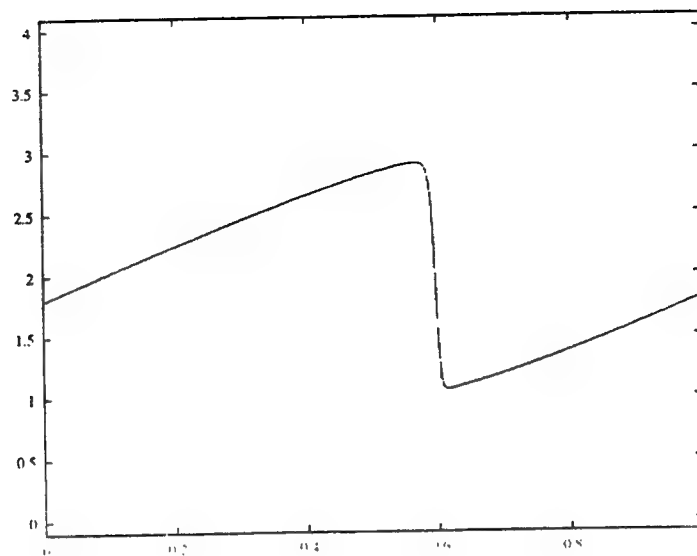


Figure 5: Problem 3 with $J_0 = 256$, $L = 5$, $\epsilon = 10^{-3}$, CFL = 0.8, and $n = 500$.

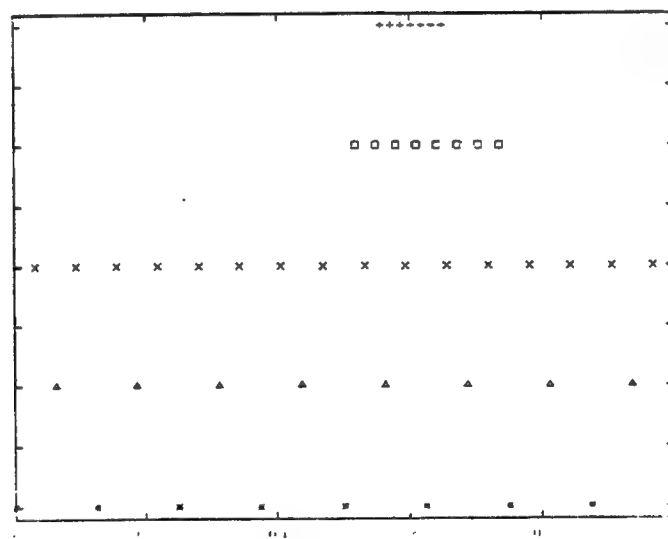


Figure 6: Problem 3: Flagged grid points

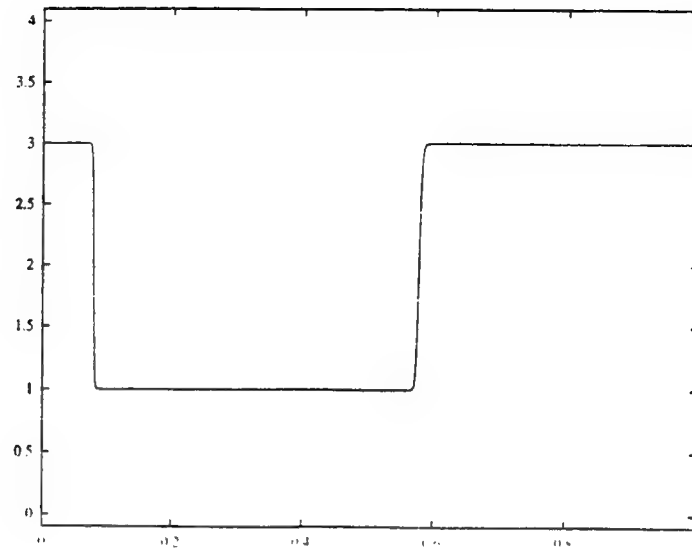


Figure 7: Problem 4 with $J_0 = 256$, $L = 5$, $\epsilon = 10^{-3}$, $\text{CFL} = 0.8$, $n = 500$, and $\mu = 500$.

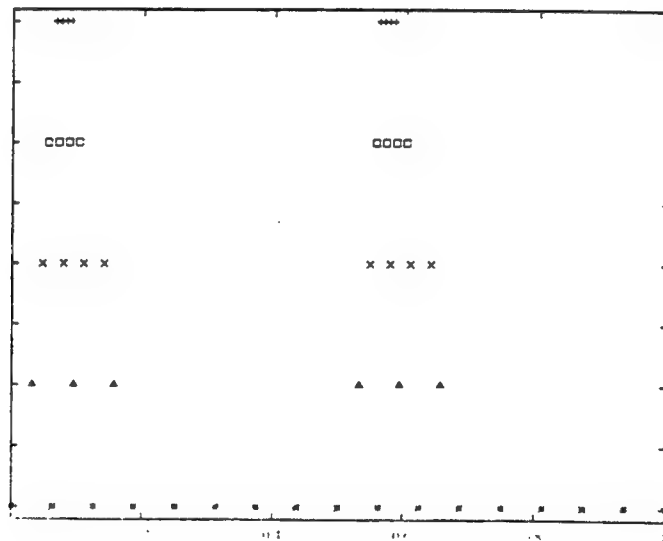


Figure 8: Problem 4: Flagged grid points

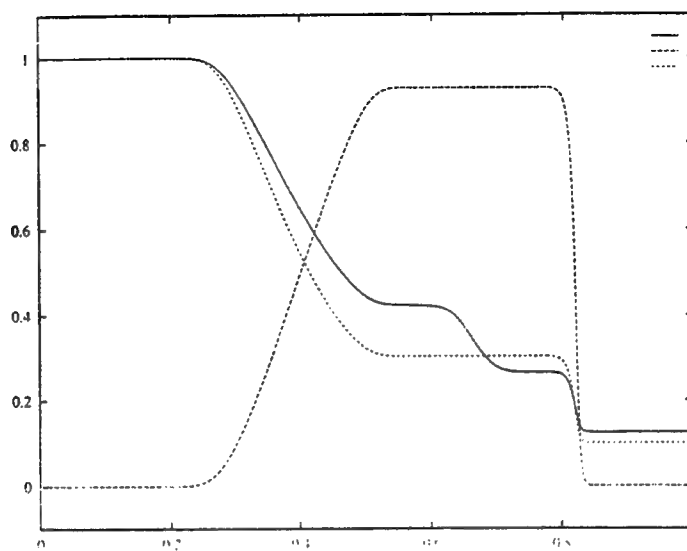


Figure 9: Problem 5 with $J_0 = 512$, $L = 5$, $\epsilon = 10^{-3}$, CFL = 0.8, and $n = 250$.

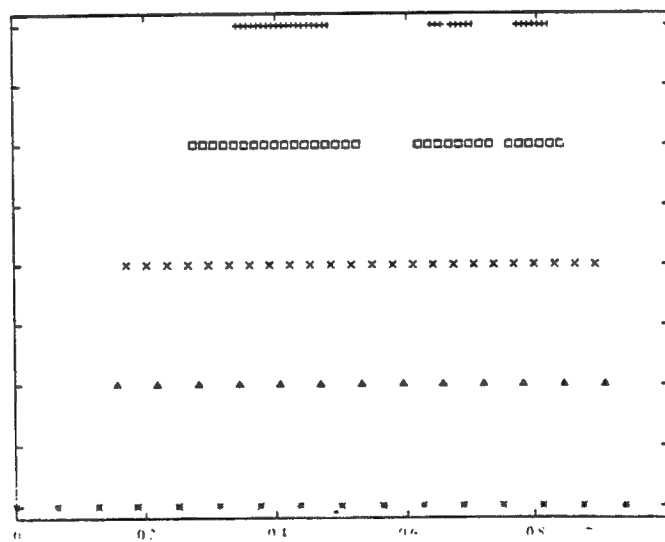


Figure 10: Problem 5: Flagged grid points

References

- [1] A. Cohen, I. Daubechies and J.-C. Feauveau, "Biorthogonal Bases of Compactly Supported Wavelets," *Commun. Pure Applied Math.* **45** (1992), 485-560.
- [2] I. Daubechies, "Orthogonal Bases of Compactly Supported Wavelets," *Commun. Pure Appl. Math.* **41** (1988), 909-996.
- [3] A. Harten, "Multiresolution Algorithms for the Numerical Solution of Hyperbolic Conservation Laws," *CAM Report 93-03* (1993).
- [4] A. Harten, "Multiresolution Representation of Data. II. General Framework," *CAM Report 94-10* (1994).
- [5] A. Harten, "Multiresolution Representation and Numerical Algorithms: A Brief Review," *ICASE Report No. 94-59* (1994).
- [6] A. Harten, "Adaptive Multiresolution Schemes for Shock Computations," *J. Comput. Phys.* **115** (1994), 319-338.
- [7] R. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser Verlag, Basel, 1990.
- [8] R. LeVeque and H. C. Yee, "A Study of Numerical Methods for Hyperbolic Conservation Laws with Stiff Source Terms," *J. Comput. Phys.* **86** (1990), 187-210.
- [9] P. L. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comput. Phys.* **43** (1981), 357-372.
- [10] H. C. Yee, "Upwind and Symmetric Shock-Capturing Schemes," *NASA Technical Memorandum 89464*, Ames Research Center (1987).

**A STUDY OF THE EFFICIENCY
OF THE CODING STRUCTURES IN THE CFD CODE XAIR**

**Robert L. Roach
Assistant Professor
Mechanical and Aerospace Engineering**

**University of Tennessee Space Institute
B.H. Goethert Parkway
Tullahoma, TN 37388-8897**

**Final Report for:
Summer Faculty Research Program
Arnold Engineering & Development Center**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC
and
Arnold Engineering & Development Center**

September 1995

A STUDY OF THE EFFICIENCY OF THE CODING STRUCTURES IN THE CFD CODE XAIR

Robert L. Roach
Assistant Professor
Mechanical and Aerospace Engineering

Abstract

The work reported here is concerned with some of the basic programming structures in the comprehensive CFD code XAIR. The code makes use of subroutines which solve block tridiagonal systems. These subroutines and the ones which load the associated arrays typically consume 70-75% of the CPU run time. Since the algorithms for these subroutines were devised some time ago, it was deemed useful to enquire as to whether changes in computer architecture over the years warranted a review of the actual coding. It was found two of the block solvers suffer from severe "striding" problems on work stations. An alternative set of block solvers was examined for speed of computation since little comparison information has been reported in the literature. The existing block solvers were the fastest on large scale computers but were the slowest on work stations due to the striding problem. However, a modification was found which produced an execution time reduction of a factor of 10. Unfortunately, large portions of the code would be required to be changed. An alternative was found for the two block solvers which still reduced the time of computation (factor of 4) and which was also transparent to the rest of the code. Subsequent testing on a realistic, multiple grid problem revealed a 21% reduction in run time with the modified routines. The work predicts an overall reduction in run time of about a factor of two if the proposed index rearrangement were to be implemented.

A STUDY OF THE EFFICIENCY OF THE CODING STRUCTURES IN THE CFD CODE XAIR

Robert L. Roach

Introduction

Computational Fluid Dynamics (CFD) at AEDC has been a major effort for many years. It not only supplies a necessary analytical capability for studying anomalous aerodynamic phenomena, it has also been used to supplement and guide wind tunnel testing. At its best, it can significantly reduce the time required to certify an aerospace system or illuminate, through a graphical presentation, flaws in the aerodynamic design of the system. As a consequence, the CFD effort at AEDC has become an integral and vital part of the mission.

A natural consequence of the routine use of CFD codes has been the identification of code limitations. These limitations are often one of slow run times due to stability limitations and large grid requirements. Whenever an improvement to a CFD tool is possible, the efficiency of the CFD computations are directly affected, and the overall process is improved by either reducing the computer resources required for a given job or by reducing the wall-clock time required for a solution. This directly improves the response time of the CFD practitioner to the customer, which is highly desirable.

The XAIR CFD code, one of the main flow solving algorithms at AEDC, uses the popular Approximate Factorization algorithm and is based on the ARC (Ames Research Code) devised at least ten years ago. The code has been appropriately adapted to the special requirements of AEDC needs and has state-of-the-art capabilities in these special needs areas such as user interfaces and CHIMERA grid embedding. While some experimentation with the basic solution procedure has been performed at AEDC, little about the basic algorithm has been changed. The original Ames code was developed to run optimally on the large CRAY Research computers of that era and little about the basic algorithm has been modified since that time.

Since the code is now run on several computer platforms, each with its own peculiarities and efficiency requirements, it seemed prudent to examine the XAIR coding to see if the execution times were still optimal. The part of the code selected for examination was the series of block solvers since these routines and associated matrix loading routines consume 70-75% of the run time. The effort here was divided into two parts:

1. Are the present block solvers the most efficient of the block solvers available? The answer to this question turned out to depend on the computer used. They were the fastest on the

larger, vector-type computers, but slowest on the workstation type. Close examination of the actual coding used in the block solver revealed that if certain modifications were devised, then the block solvers would, in fact, be the fastest on virtually all computers. This led to the second task.

2. Examine the required modifications for improved efficiency (fastest run time). Here, the solvers and associated matrix filling routines were to be examined to determine more optimal code structures for faster run times. The best way to modify the routines turned out to hinge on rearranging the 4 indices of the large coefficient matrix arrays. Unfortunately, this would involve a considerable amount of work and testing since the arrays appear in about 24 different subroutines and each would have to be modified (including the loop structures) and then validated. An alternative was found which could be implemented in the two offending block solvers without having to modify the remaining subroutines. This modification was performed, the routines validated, and a realistic, 3D, multiple grid problem was solved. The run times were compared to those of the same problem using the original version of the XAIR code. An overall run time reduction of 21% was achieved with the modified block solvers.

This latter step made a significant step toward reducing the computer run time. It is recommended, though, that a more comprehensive set of changes be made to the XAIR code based on the original idea of rearranging the indices and modifying the associated subroutines. It is estimated that the overall run time reduction would be a factor of 2 for workstations and about 20% on the large scale computers.

The remainder of this report summarizes the two steps above. In the following section is a list of programming efficiency principles as is applicable to the XAIR code. This is followed by the effort in comparing the run time speed of several different block solvers. The next section describes a few of the coding issues which affect run time speed and the types of coding structures found in the XAIR block solvers. Finally, an implementation of a fix to two of the block solvers which leaves the remaining code untouched is described, followed by a recommendation for a more comprehensive and complete recoding.

Some Program Efficiency Principles in Large CFD Codes

Most large computer codes that numerically solve systems of partial differential equations have similar coding structures. There are a number of practices which should be avoided if they are to be performed a large number of times (> 100,000, say) since they can

considerable affect the amount of computer run time. Instead, alternative coding structures should be used. Examples of the ones relevant to the XAIR code are as follows:

1. Avoiding "Stride" problems by insuring that the leftmost index in multidimensional arrays corresponds to the innermost DO loop since the leftmost index corresponds to the fastest incrementing index. This is not always possible. However, there are many times when a multiple DO loop may be put in any order and the above should be implemented. The reason is that most workstations, in executing a loop with arrays, will assume a stride of one and extract from memory the next variable needed in the array along with a chunk of the neighboring values also in that array. If the loop index does not correspond to the leftmost index of the array, then the next value needed will be much further away in memory and will not have been extracted along with the last fetch. This will result in another expensive (timewise) memory fetch for each increment in the innermost loop. The memory fetch would have been avoided, if the loops were arranged as recommended. Larger computers with a large cache memory can avoid the problem. In addition, some computers have compilers which allow strides larger than one in the information that is fetched. Both cases avoid the problem, but most workstations have neither feature. This is problem is the one which most affects the block solvers in XAIR.

2 - Multiplying by an inverse rather than dividing. Table I presents the run times required by certain arithmetic operations as well as calls to some intrinsic functions. It can be seen that the division operation is 50% slower than a multiply operation (800% on the IBM RS 6000 machine). Thus, if a loop has division by a constant, it would be more economical to define the inverse of the constant and multiply by it. Another example of this includes multiplying by 0.5 instead of dividing by 2.0.

Table I. CPU time (μ sec) per operation.

Operation	SGI Indigo 2	IBM RS6000	HP 735
+	.100	.033	.040
-	.100	.033	.040
*	.112	.033	.041
/	.183	.251	.064
$\sqrt{\quad}$	1.086		
log	1.208	.906	.692
exp	1.184	.879	.647
sin	.842	.482	.814
cos	1.093	.492	.790
tan	.980	1.021	1.451

3. Multiplication in a loop by more than one constant. In this case it is more economical to define the product of the constants outside the loop and multiply by the product inside the

loop. While some compilers will automatically do this, others will not and it is good programming practice to do this.

4. Reducing the operation count in inefficient loops. Because many statements in a computer program are direct copies of formulas, excess, or redundant computations are often done inadvertently. Significant reductions in operation count can be obtained with a corresponding reduction in run time if these loops were merely rewritten to remove the redundancy.

Incidences of these four items were found in several places in an otherwise well-written XAIR code. In particular, the stride problem played a major role in creating the large run times in two of the block solvers. The first effort, then involved an inquiry as to the run time comparison of several other block solvers and whether one of them should be used in XAIR.

Examination of a few Block Tridiagonal Solvers

Several block tridiagonal solvers are available at AEDC and UTSI. To compare the computational speed, a test program was written with each of the routines added as callable subroutines simulating the manner in which they are normally called. Two additional block solvers were written which were simple modifications of the ones tested. These were also placed into the test program. The various solvers are described next followed by a description of the test problem. The timing results are given in Table II.

A. The Solvers

The block solvers available at AEDC and UTSI include those described below. While there are others available, the ones tested here are considered representative. The main differences in the codes involved the structure of the solution method. Most are built around the LU scheme described by Isaacson and Keller [1]. Other differences include two solvers which assume a specific block size of 5 (since that is the block size for the three-dimensional Navier-Stokes equations). When the block size is assumed, greater program efficiency can be obtained by using an "unwrapping" structure. Another difference is that some of the solvers were set up to solve multiple systems of block tridiagonal systems at one time. This is useful on vector machines where the vector length of a single system is extremely short and vectorization gives worse run times. The multiple system solution allows vectors lengths equal to the number of systems being solved.

BTRI5 - this routine is one of series written by Greg Molvick [2] and uses an unwrapped LU decomposition for a specific block size. If the block size changes, then a different version of the series must be used. It also solves a single block tridiagonal system at a time. This code was by far the fastest on the scalar architecture machines and as fast as BVSOLV on the vector machine.

BTRIR- this is a version of BTRI5 written by Prof. RL Roach which eliminates the restriction on the single block size. This was done by recombining the statements into DO loops. This version was 2-3 times slower than the unwrapped BTRI5 and about the same speed as BSOLV.

BSOLV - this routine, written by Prof. RL Roach [3] solves a single block tridiagonal system at a time. It was optimized for CDC Cyber 175 architectures at the time it was written. It is written to be independent of the direction of the tridiagonal system with respect to the grid.

BVSOLV - this is the vectorized version of BVSOLV written by Prof. RL Roach [4] and solves a series of block tridiagonal systems at one time. The index of the systems was placed in the leftmost position of the coefficient matrices and is also the innermost loop index in the solver.

BTRIX - this is one of the block solvers that exists in XAIR. It was written for a series of systems and apparently optimized for vector machines. BTRIX is one of a series of solvers which have been written to be associated with certain directions in the grid. BTRIX was the fastest solver on the Convex when the vector and vector/parallel options were invoked. It was the slowest on the scalar machines.

B. The Test Problem

A 2-D problem was set up which could be called a number of times to simulate the 3rd dimension. This is shown in Fig. 1. The tridiagonal systems were aligned in the i-direction and the second dimension was in the j-direction. The grid was 101x75 and the number of cycles (simulating a third direction) was usually set at 10. The coefficient matrices for the test were

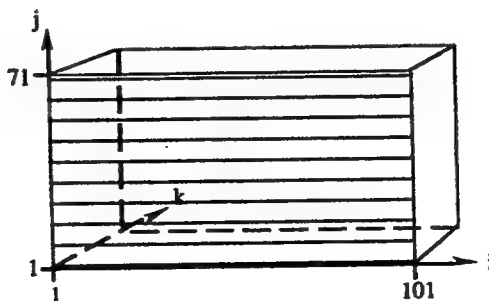


Fig. 1. Geometry of Test Problem.

computed in the following manner:

$$\begin{aligned}A_{ij} &= C_{ij} = \frac{-1}{1 + |i - j|} \\ B_{ij} &= 1 - (A_{ij} + C_{ij}) \\ D_j &= j\end{aligned}$$

where:

$$i \in [1,5] \quad j \in [1,5]$$

This definition gives a symmetric and strictly diagonally dominant set of matrices. Every matrix was the same throughout the domain. The solution was checked at a single point. When properly invoked, each of the solvers gave the same answer for the 5 dependent variables at that point to 4 decimal places when double precision was used. Significantly, if single precision was used, there were differences in the 3rd decimal place. BSOLV and BVSOLV gave single precision solutions closest to the double precision solutions.

C. The Computers and Compilers

Four computers were used to perform the timings, an IBM RS-6000 based machine (ibm00), a Silicon Graphics IRIS workstation (troi), both of which are scalar machines, a Convex, and an SGI Power Challenge-L 2-processor system. The latter two are capable of vector and parallel computation.

Each machine has its own compiler with various compilation options. These options can significantly change the timing results so the timings are presented for the most significant cases. Timings were done in both single and double precision. However, since the single precision solutions are suspect, they are not reported in this note. The differences in run times between single and double precision were not particularly large, being, at most, about 8%.

The timing was done by using the intrinsic timing functions available for the compiler. On the IBM and SGI machines a generic timer named MCLOCK reports the hundredths seconds elapsed since midnight. MCLOCK was called at the start of a routine and at the end of each solution procedure and the difference is divided by 100 to get the CPU seconds. On the convex, there is a "profiler" which examines a performance output file which indicates the amount of time spent in each subroutine.

The reported run times are an average over several runs since the time can vary in any given run due to the number of swap-outs, internal interrupts, and the like. However, the variation is not particularly great, being, at most about 5%. Thus the reported times are not a guarantee but are typical.

D. Results

The timing results shown in Table II are given in CPU seconds for a run using the geometry indicated previously. The run times include the time required to load the coefficient matrices and the time the solver uses to produce the solution. One difference in the calling routines is that the solvers BVSOLV and BTRIX only needed to be called once for each plane (in this case 10 times), whereas the others were called once for each line (710 calls).

Table II. CPU Run times for the test problem (CPU sec)

Computer	Compiler	Options	BTRI5#	BTRIR	BSOLV	BVSOLV	BTRIX#
IBM RS600	xlf	none	3.99	12.56	12.25	17.50	11.17
		-O	.80	1.75	2.23	4.87	8.92
		-O -Q	.81	1.78	2.38	4.45	8.59
		-O3	.81	1.92	2.40	4.66	8.96
		-O -qarch=pwrx	.81	1.82	2.23	5.02	8.72
SGI Indigo 2	f77 (troi)	none	4.87	12.85	9.79	20.65	14.97
		-O2	2.65	5.29	4.65	11.09	11.87
		-O3	2.62	5.29	4.64	11.28	10.87
Convex	fc	none	10.52	27.20	32.78	43.18	28.16
		-O1	4.03	12.53	13.26	11.26	11.67
		-O2*	3.36	18.07	13.93	3.36	2.51
		-O3**	4.79	47.80	40.95	5.25	1.94
SGI Pwr Ch.	f77	none	6.86	15.00	16.65	24.53	10.39
		-O2**	1.73	3.40	3.56	4.26	2.12
		-O3**	1.96	3.48	3.76	3.92	1.67

* Vector option

** Vector and parallel option

hard coded for block size of 5

The results show that the block solver used in the code XAIR possess the smallest run times in vector and vector/parallel modes on the Convex computer. However, it has the slowest times of those tested on scalar work stations without a vector or vector/parallel capability. Two single system block solvers, BTRI5 and BSOLV, were considerably faster (up to a factor of 10) on the scalar workstations. To get the best performance on all machines one could conclude that two versions of XAIR should be written: one with the original block solver BTRIX for use on parallel machines and another for scalar machines with BTRI5. This was deemed an unsatisfactory solution since this could open the door for multiple versions of the code and the subsequent difficulty of keeping up with all the versions.

The other solution for optimal performance on all machines was to examine the BTRIX subroutine to see if some modifications might allow for a more optimal operation on scalar machines. The next section describes the successful efforts at optimization of BTRIX.

Optimization of the BTRIX Subroutine

Since the BTRIX subroutine is the main block solver used by the XAIR code, its slowness on the scalar workstations was examined and primarily determined to be caused by the excessive striding in the loop structure. Striding can be caused by several factors, not the least of which is the use of innermost loops which do not correspond to the leftmost index of the quantities being used. This results in stepping through memory with increments larger than one. The increment is also called the stride. As a consequence, larger cache sizes are needed to remain efficient. In BTRIX, the stride or loop increment amounted to the size of the first three indices, which is a large number. To fix the stride problem in BTRIX requires a change to some of the loop structures and to the order of the indices in the 4 main arrays, A, B, C, and S.

To understand the arrangement of indices, it may be recalled that BTRIX is simultaneously solving a series of block tridiagonal systems. Each system consists of a block matrix with three main diagonals. Each row of the matrix thus has three elements, A, B, and C, which are each a 5x5 matrix. The right hand side, S, is a vector with elements which are 1x5 matrices. Each row of this block matrix corresponds to the finite difference representation of the Navier-Stokes equations at one grid point in the domain. Thus the block tridiagonal system corresponds to solving for the flow quantities along an entire row of grid points. Fig. 2 shows a single block tridiagonal system. Three of the indices used in A, B, and C are shown in the figure. An index, j, represents the row of the matrix and the indices, m and n, represent row and column of the 5x5 matrix. A 4th index, l, in A, B, and C corresponds to a specific block system.

In the XAIR code the arrangement of these indices are: A(j,l,m,n). However, this is not the most advantageous order of indices. The most advantageous arrangement of indices arises from an analysis of the order of operations in the vectorized version of solving the systems. Consider first the solution of a single block tridiagonal system. In that process operations are required on the individual 5x5 matrices. As a consequence, an outer loop on each row of the system has inner loops of lengths 5 or fewer. Thus the innermost loops in such a process are quite short and correspond to row and column operations, i.e. on the m and n indices. The outer loop (actually there are several outer loops) uses the system row index j. Thus, it should be clear from this that the row index j should be placed last in the specification of the arrays A, B, and C since it corresponds to outer loops.

With short inner loops, the block matrix solver does not vectorize well. To overcome this, a large number of these systems is solved together. A new index, l , is used to distinguish between the systems. The block solving subroutine is thus "vectorized" by performing each operation of the solving process on each of the l systems using a new innermost loop on l . This clearly indicates that l should be the leftmost index. Hence an optimal arrangement of indices for the coefficient matrices is: $A(l,m,n,j)$ or $A(l,n,m,j)$. No real studies have been performed on the difference between these two arrangements, however, the arrangement of the j and l indices is the most critical. In the original version of XAIR it should be noticed that j was placed first in stead of last. This is the cause of the stride problem.

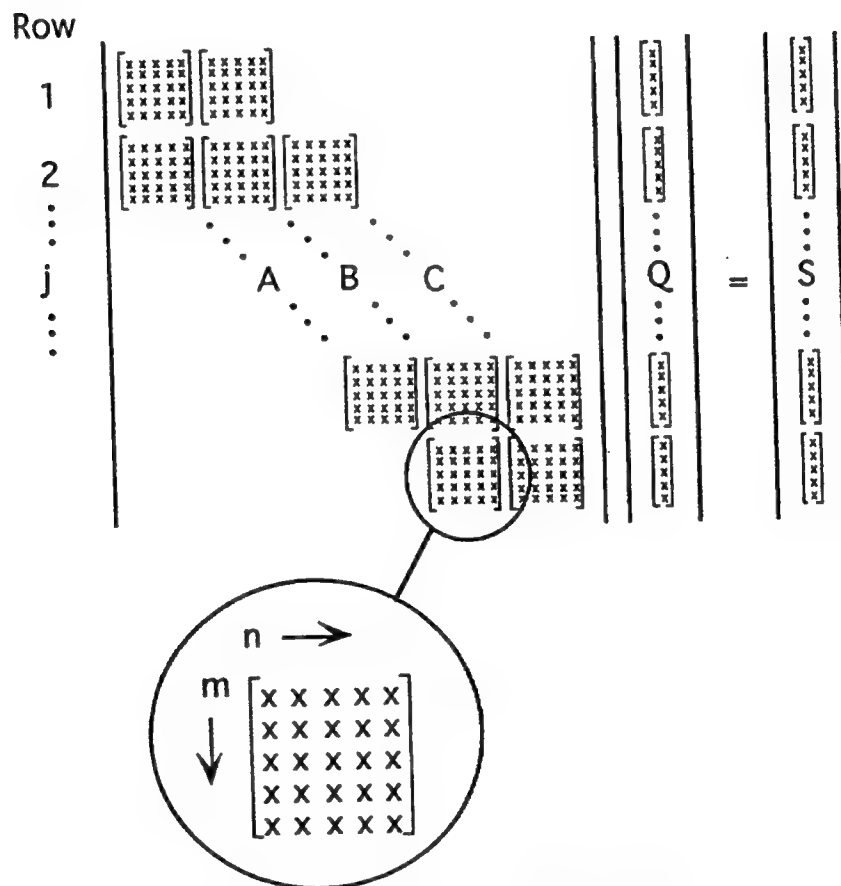


Fig. 2. Single Block Tridiagonal System.

To fix this would require the rewrite of all of the subroutines which use the matrices A , B , and C . These include the setup subroutines, the ones which load the coefficient matrices and the block solvers. Such an effort would be worth it if the run time savings is large. To check this a less comprehensive experiment was performed. First, a modified version of the BTRIX subroutine was written, called BTRXM, where the indices of A , B , and C were arranged as was

deemed optimal. Next BTRXM was applied to the model problem given in the original timing studies. The results show a significant speed up over the original version. On the IBM RS6000 machine the speed-up is by a factor of 11 making BTRXM the fastest on that machine. The speed-up on the SGI workstation was a factor of 2.5 and a factor of 1.2 on the Convex.

These results are good enough to warrant a further look at the other routines which load the matrices to see the impact of rearranging the indices. Since such an effort is beyond the scope of the present study a simpler version of BTRXM was devised which did not require the modification of any other routine in XAIR. In this version, new matrices, AA, BB, and CC were created which have the optimal index arrangement. When the A, B, and C arrays are passed to BTRXM, they are put into AA, BB, and CC with a series of loops. BTRXM then carries out the solving process as before. The same test as above was run. The IBM RS6000 machine found a speed-up of a factor of 4.

To see the effect on a realistic problem, the modified subroutine BTRXM was in the original code in place of BTRIX. The following section describes the test.

Test of the BTRXM Subroutine in a Realistic CFD Problem¹

A typical realistic CFD problem involved the computation of the viscous ($Re=2.4 \times 10^6/ft$) flow over a wing-pylon-store configuration at a Mach number of .95 shown in Fig. 3. The configuration is upside down to simulate the separation tests performed in AEDC wind tunnels. Eleven grids were used with a total of about 1.38 million grid points. The steady state computation of such a configuration on the IBM RS6000 workstation requires about 120 CPU hours. A typical unsteady computation involving the separation of stores from the wing would consume about 3000 CPU hours. Thus any decrease in run time due to an improvement in run time efficiency has a tremendous impact.

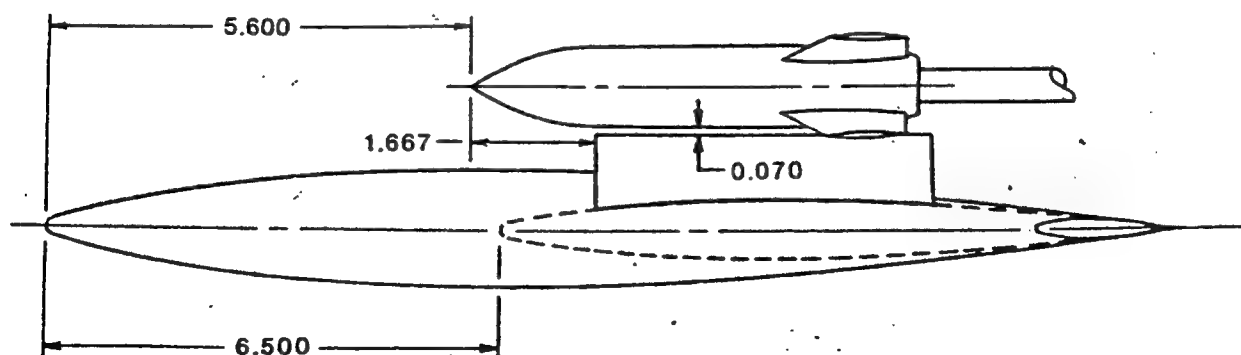


Fig. 3. Wing-Pylon-Store (with fins) Configuration.

¹This work was done by Norm Suhs who was kind enough to let me interject this test on the important runs he was conducting.

Use of the BTRXM subroutine in place of the BTRIX subroutine for one steady problem resulted in a 21% decrease in the run time. All of the decrease was attributed to the lower run time required by the block solver. The other block solvers, BTRIY and BTRIZ were not modified since their strides have less impact. However, changing the indices in these routines are estimated to be able to result in additional overall run time decreases for a typical run of 5-10%. In addition, the subroutines which load the A, B, and C arrays could be rewritten in a more efficient manner in both the index rearrangement and in removing coding redundancies.

Conclusions

The work in this project involved trying to determine if the coding structures in the XAIR code were optimal for the types of computers on which the code is executed. It was determined that the main routines which consume 70-75% of the run time have very large arrays with a non-optimal index ordering. The runtime gain on the costlier routines can be as high as a factor of 11 on some computers if an optimal arrangement is used. This not only would allow the same block solver to be used across all platforms, it would also result in an overall run time reduction on all platforms.

An examination of the matrix loading subroutines has revealed that, in addition to the non-optimal arrangement of the indices of the A, B, and C arrays, there are some other coding structures which violate the other principles outlined above. These include multiplication by multiple constants as well as one set of loops in which redundant computations are performed. Since this pattern is repeated in all of the matrix loading routines, there is a distinct possibility of further run time reductions in these routines by simply reducing the operation count.

If all of the changes were to be implemented into XAIR, it should be possible to lower the over computer run time by as much as a factor of two. Since several large jobs count computer run in man-weeks or even man-months, this represents a considerable savings.

References

1. Isaacson, E. and H.B. Keller, Analysis of Numerical Methods, Wiley - NY, 1966.
2. Molvik, G.A. and C.L. Merkle, "A Set of Strongly-Coupled Upwind Algorithms for Computing Flows in Chemical Nonequilibrium," AIAA 89-0199, 27th AIAA Aerospace Sciences Meeting, Reno NV, Jan 1989.
3. Roach, R.L., "An Implicit Finite Difference Procedure for the Laminar Supersonic Base Flow," PhD Thesis, Georgia Institute of Technology, Dec 1977.
4. Roach, R.L., J. Jenkins, and S. Lash, "Application of the Unsteady, Three Dimensional Navier-Stokes Equations to Sudden Pitch Change in the High Speed, High Angle of Attack Delta Wing," AIAA-89-0646, AIAA 27th Aerospace Sciences Meeting, Reno NV, Jan. 1989.

**NON-LINEAR RESPONSE
OF AN INTENSIFIED CHARGE-COUPLED DEVICE (ICCD)
CAMERA TO NANOSECOND LASER PULSES**

**John T. Tarvin
Associate Professor
Department of Physics and Astronomy**

**Samford University
800 Lakeshore Drive
Birmingham, AL 35229**

**Final Report for:
Summer Faculty Research Program
Arnold Engineering Development Center**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington, D.C.**

and

Arnold Engineering Development Center

September 1995

NON-LINEAR RESPONSE
OF AN INTENSIFIED CHARGE-COUPLED DEVICE (ICCD)
CAMERA TO NANOSECOND LASER PULSES

John T. Tarvin
Associate Professor
Department of Physics and Astronomy

Abstract

Dual-wavelength planar laser-induced fluorescence (PLIF) measurements of shock flows provide a non-invasive technique for the visualization of temperature profiles within such flows. These measurements typically use an intensified charge-coupled device (ICCD) camera to record ultra-violet (uv) fluorescence images excited by a Nd:YAG/dye laser system. The response of this camera system requires "flat-fielding" of the images due to significant variations of the camera response function across the field of view. It has been determined that the commercially-obtained cameras in the PLIF system used for the present study are non-linear in their response. This non-linearity is most pronounced over the upper 75% of the dynamic range of these cameras; camera response can be considered essentially linear only over the lower 20-25% of the camera range. This non-linearity is only present when camera images are obtained using laser pulses (order of 10 ns) as an illumination source; continuous wave (cw) illumination with a deuterium lamp source provides a linear response over the full dynamic range of the ICCD camera. All images were obtained using uniform illumination from a Lambertian plate. All camera variations across the image are correctable using standard flat-fielding techniques; the resulting correction image is independent of the illumination source provided the camera is operated within its linear response region.

NON-LINEAR RESPONSE OF AN INTENSIFIED CHARGE-COUPLED DEVICE (ICCD) CAMERA TO NANOSECOND LASER PULSES

John T. Tarvin

INTRODUCTION

Planar laser-induced fluorescence (PLIF) is a well-established technique for the non-invasive visualization of flowfields (1-6). When used at two wavelengths, this technique allows visualization of flowfield temperature (4,6). A fluorescence image is usually collected through the use of an intensified charge-coupled device (ICCD) camera (3-5). Consequently, the response of the ICCD camera is of key importance to the accurate interpretation of the PLIF measurement. Previous work in this area has investigated the response of an ICCD camera system in the presence of both continuous illumination (7) and pulsed illumination (8). Surprisingly, the pulsed-illumination work indicated a non-linear response of the ICCD camera over the upper 75-80% of its range (8). Since actual PLIF measurements use a short laser pulse (4) questions of possible camera response differences between continuous and pulsed illumination and their effects on image interpretation are relevant. One of the cameras used in the previous study was returned to the manufacturer for a linearity check. The manufacturer concluded that there was, in fact, no linearity problem with this camera (9). As a result of this seeming contradiction, a more detailed examination of this effect has been conducted.

APPARATUS AND METHODOLOGY

The PLIF measurement system has been described previously (3,4). The primary difference

between the present make-up of this system and that reported previously is the replacement of the PG-10 high-voltage pulser unit with the newer PG-50 unit (both by Princeton Instruments). The control program, CSMA (Princeton Instruments), has also been updated to version 2.3.

All reported measurements are for a lens-less ICCD camera system. Flat-field illumination was provided by illuminating a Lambertian plate (Labsphere Spectralon Diffuse Reflectance Target) with either a deuterium lamp (Hamamatsu) or a pulsed YAG laser. The deuterium source was mounted a fixed distance from the plate, and both items were then mounted on a sliding track. Intensity variations were obtained by varying the distance of the plate from the fixed camera. The camera was shielded from direct illumination by the source. Laser illumination of the plate was obtained using the same axis as the deuterium lamp. Laser intensity was varied by attenuating the beam with an iris. The setup is shown schematically in Figure 1. All measurements were taken using the same camera filters that are used in the PLIF measurements; these filters provided a narrow passband centered near 248 nm.

Laser pulse energy levels were monitored with a PIN diode (Thorlabs). Previous calibration of this diode with an energy meter indicated that its response was a linear measure of the beam energy (8). The deuterium source energy variation was assumed to follow an inverse square law. Since an extended source was being used, this assumption was tested in two ways. First, the same PIN diode was used to measure the energy (voltage) variation as a function of distance. This measurement used a chopper and a lock-in amplifier in order to minimize diode drift with constant illumination. The measured variation was reasonably linear, especially over the range of distances used in the camera study. A computer simulation was also performed. In this case, the plate was broken up into a 100x100 grid, and the individual sections were treated as point sources. These sources were then

summed to give the actual intensity due to the extended source as a function of plate distance. The resulting values varied less than 1 % across the intensifier face (1 inch diameter), and the intensity variation varied in an inverse square manner to better than 3 %. Consequently, distance from the plate to the intensifier surface was used to calculate a value proportion to the actual intensity. Preset scale readings corresponding to intensity levels of 100 %, 90 %, 80 %, etc., were determined; images were then taken at these distances. An initial distance was chosen such that the camera was near saturation for the 100 % distance.

Data collection required an intensifier gate width of 5-10 ms for the deuterium lamp source in order to get camera saturation. All YAG data was collected for a gate width of 5 ms, even though the pulse itself was only on the order of 10 ns or less. Data was collected with the camera both unblocked and blocked with a neutral density (ND) filter. If the camera responds linearly with energy, the two data sets should be the same to within a constant scale factor.

RESULTS

Measured PIN diode voltage as a function of distance from the diode is shown in Figure 2. The solid line represents a quadratic fit to the data, and indicates that there is a small, but significant, downward concavity to the data. Although a distance measurement error of about one centimeter could straighten this calibration curve, it is doubtful that the measured distances contain this much error. Rather, the curvature is most likely real, and represents possible non-linearities in the diode, coupled with deviations from an exact inverse-square relationship due to stray light and/or the use of an extended source. Because of this uncertainty, and because of the relatively small amount of curvature present, the following data will be plotted as a function of inverse distance squared, rather than as diode voltage. In most cases, the data noise obscures such corrections.

Data were taken for various intensifier gains of the camera. Selected gains were consistent with those gains actually used for PLIF work. Camera response was consistent with respect to this gain variation, in the sense that the pixel-to-pixel ratio of two images obtained at two separate gain settings resulted in a constant image (to within the noise of the individual pixels). Data replicates were also collected over several days. Again, camera response was consistent among these several data sets, suggesting that the measured camera responses are not anomalous.

A typical set of responses for cw illumination (deuterium lamp) are shown in Figure 3. The data points represent averages of a 100x100 pixel region in the "flattest" part of the image. The error bars represent the standard deviation within this region. It can be seen that the two curves are reasonably linear over their entire range; the solid line is a linear fit to the background-corrected data, constrained to have a zero intercept. The base-ten logarithm of the slope ratio for these curves is 0.47, which is within about 5 % of the measured value of 0.45 (at 250 nm) for the ND filter used. Based on findings such as these, it is concluded that the ICCD camera responds linearly over its dynamic range under cw illumination. This conclusion applies to each of the three cameras currently being used for PLIF work at Arnold Engineering Development Center (AEDC).

The above measurements were repeated with the same physical setup, except for three differences: 1) the Lambertian plate remained at a fixed distance from the camera, 2) beam energy incident on the plate was varied by varying the diameter of the laser beam using a circular iris, and 3) beam energy was measured as the maximum pulse height from a PIN diode that was viewing the plate from a mount beside the camera. The result of these measurements is shown in Figure 4. This data represents two unblocked replicates taken on separate days and one blocked data set; all geometrical factors are constant for these data sets. It can be seen that the system was very

reproducible, as evidenced by the overlap of the two unblocked data sets. It is also evident that the blocked versus unblocked data sets differ by more than a multiplicative scale factor over the upper dynamic range of the camera. Figure 5 is an expansion of the lower, linear part of the data; the straight line is a constrained linear fit. The log-ratio of these slopes is within about 1 % of the measured ND value (0.889 vs. 0.88 measured), suggesting that the response over the lower part of the camera's dynamic range is consistent with that expected based on an assumption of linearity.

CONCLUSION

The probable source of the ICCD camera non-linearity is the intensifier array. It is known that micro-channel plate intensifiers can exhibit saturation and non-linear effects (10,11); conversely, unintensified CCDs do not appear to exhibit such noticeable amounts of non-linearity. Consequently, the combination of intense energy released over a brief interval appears to be the culprit. Since it is unlikely that this effect is correctable (see reference 6 contained in current reference 11), one should exercise caution when using the full dynamic range of the currently studied ICCD cameras. For current PLIF work at AEDC, the PLIF signal in the flowfield region of interest is less than 20% of the maximum digitization range of the ICCD camera (12); consequently, a linear correction over the lower 20% of the signal range is reasonable and acceptable. This is based on the observed fact that, even at maximum intensifier gain, linear flat-fielding over this range results in a maximum image variation of 10% or less (8).

REFERENCES

1. McMillin, B.K., Seitzman, J.M. and Hanson, R.K., "Comparison of NO and OH PLIF Temperature Measurements in a SCRAMJET Model Flowfield", AIAA Paper 93-2035, AIAA/SAE/ASME/ASEE 29th Joint Propulsion Conference, Monterey, CA, June 28-30, 1993.
2. McMillin, B.K., Palmer, J.L. and Hanson, R.K., "Two-Dimensional Temperature Measurements of Shock Tube Flows Using Planar Laser-Induced Fluorescence Imaging of Nitric Oxide", AIAA Paper 91-1670, AIAA 22th Fluid Dynamics, Plasma Dynamics & Lasers Conference, Honolulu, HI, June 24-26, 1991.
3. Havener, A.G., and Smith, M.S., "Holographic and PLIF Measurements of Free-flight Hypervelocity Flows in the AEDC Range G Facility", AIAA Paper 92-3935, AIAA 17th Aerospace Ground Testing Conference, Nashville, TN, July 6-8, 1992.
4. Smith, M.S, Williams, W.D., Price, L.L., and Jones, J.H., "Shocktube Planar Laser Induced Fluorescence Measurements in Support of the AEDC Impulse Facility", AIAA Paper 94-2649, AIAA 18th Aerospace Ground Testing Conference, Colorado Springs, CO, June 20-23, 1994.
5. Ruyten, W., Heltsley, F., and Williams, W.D., "Computational Flow Imaging for Planar Laser Induced Fluorescence Applications (CFI-PLIF)", AIAA Paper 94-2621, AIAA 18th Aerospace Ground Testing Conference, Colorado Springs, CO, June 20-23, 1994.

6. Palmer, J.L., McMillin, B.K., and Hanson, R.K., "Planar Laser-Induced Fluorescence Imaging of Velocity and Temperature in Shock Tunnel Free Jet Flow", AIAA Paper 92-0762, AIAA 30th Aerospace Sciences Meeting, Reno, NV, January 6-9, 1992.
7. Tarvin, J.T., "Characterization of an Intensified Charge-Coupled Device (ICCD) camera used in Planar Laser-Induced Fluorescence (PLIF) Studies", Final Report for Summer Faculty Research Program, Arnold Engineering Development Center, August, 1993.
8. Tarvin, J.T., "Ultraviolet flat-field response of an Intensified Charge-Coupled Device (ICCD) camera to nanosecond laser pulses", Final Report for Summer Faculty Research Program, Arnold Engineering Development Center, September, 1994.
9. Smith, M.S., private communication.
10. Eberhardt, E.H., "An operation model for microchannel plate devices", IEEE Trans. Nuc. Sci. NS-28, 712 (1981).
11. Paul, P.H., "The Application of Intensified Array Detectors to Quantitative Planar Laser-Induced Fluorescence Imaging", AIAA Paper 91-2315, AIAA/SAE/ASME/ASEE 27th Joint Propulsion Conference, Sacramento, CA, June 24-26, 1991.
12. Ruyten, W., private communication

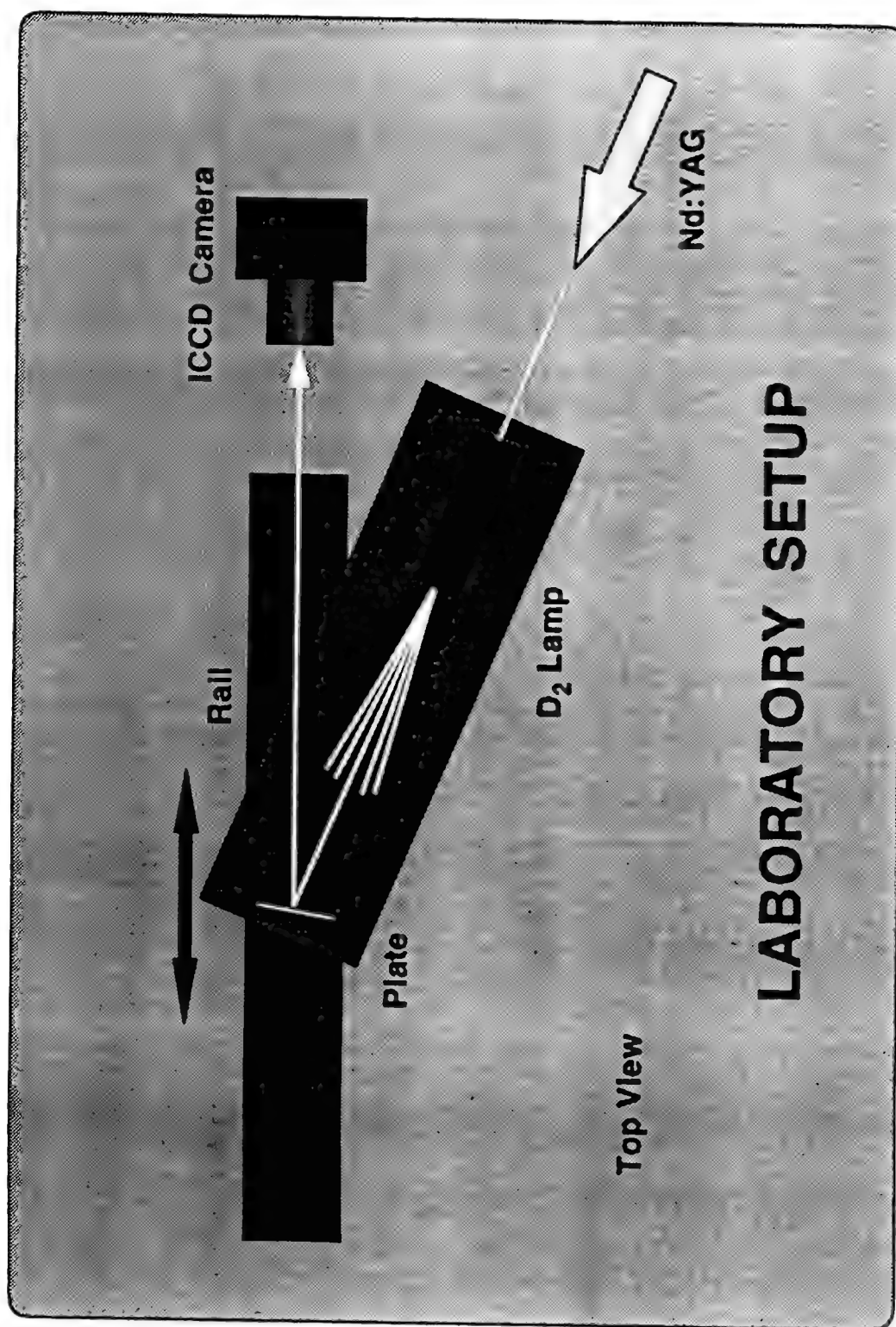


Figure 1

**Diode Voltage Measurement
(D2 lamp with lock-in amp)**

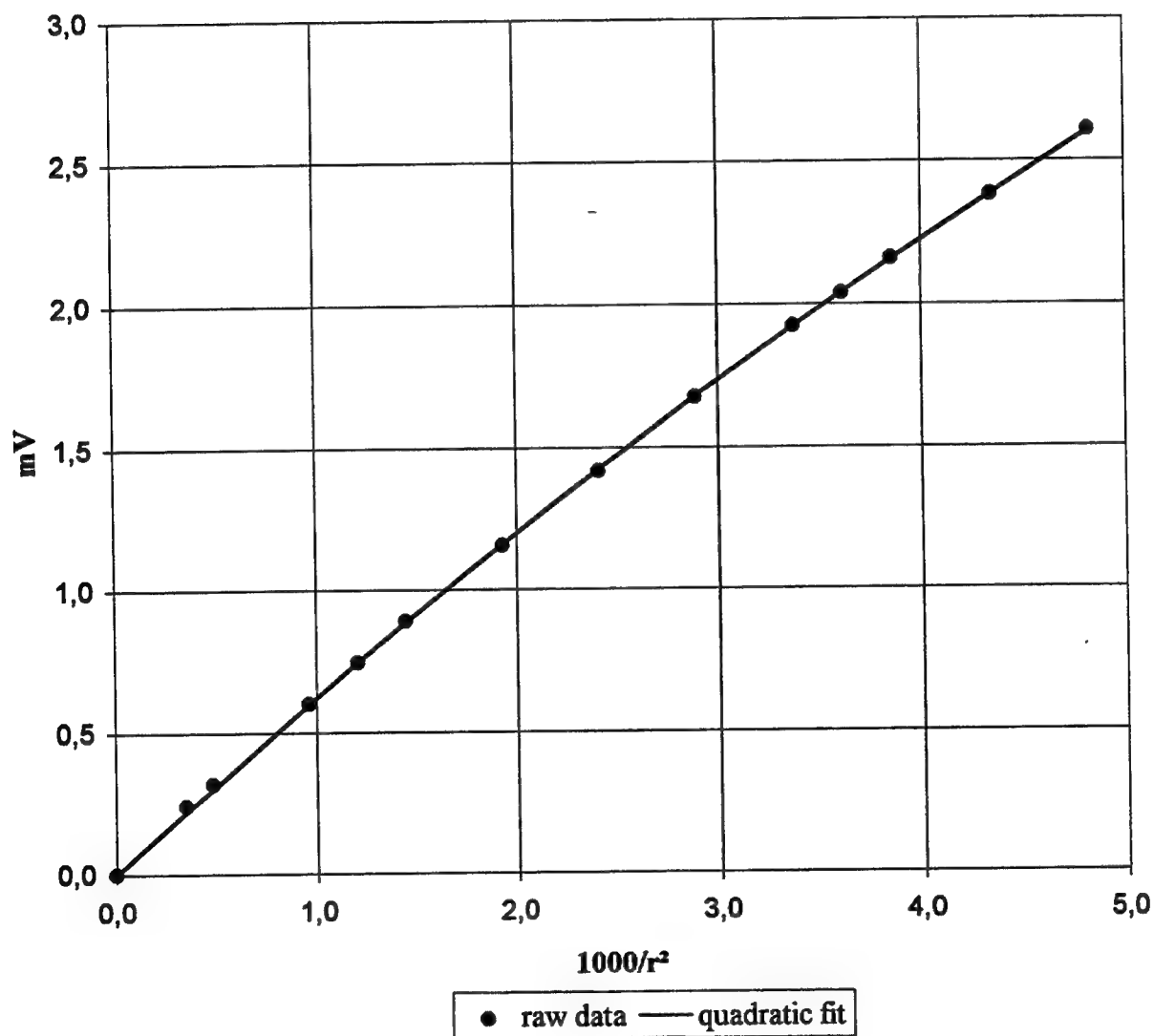


Figure 2

Camera 2 Linearity: Deuterium Source

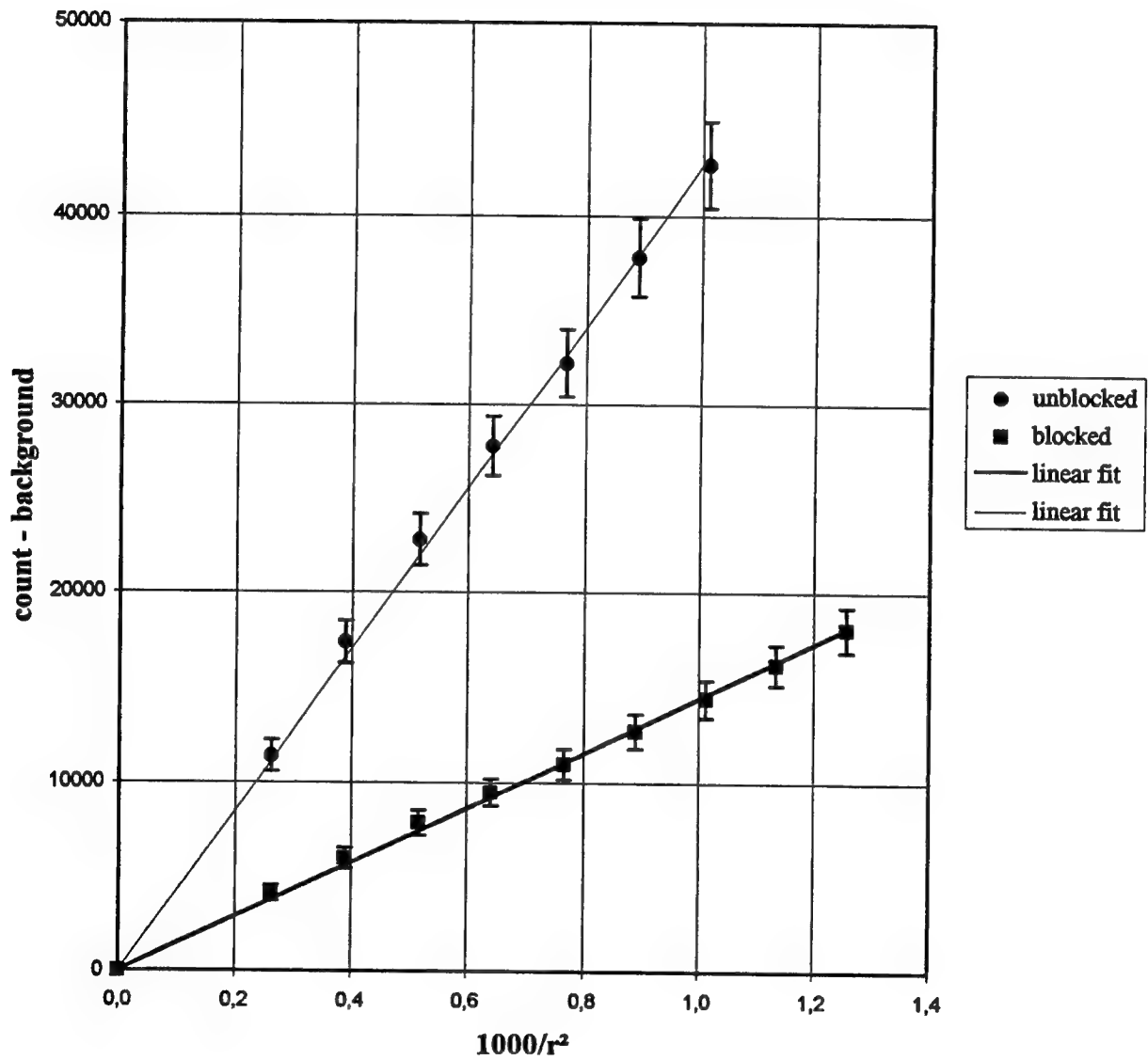


Figure 3

Camera 2 Linearity: YAG Source

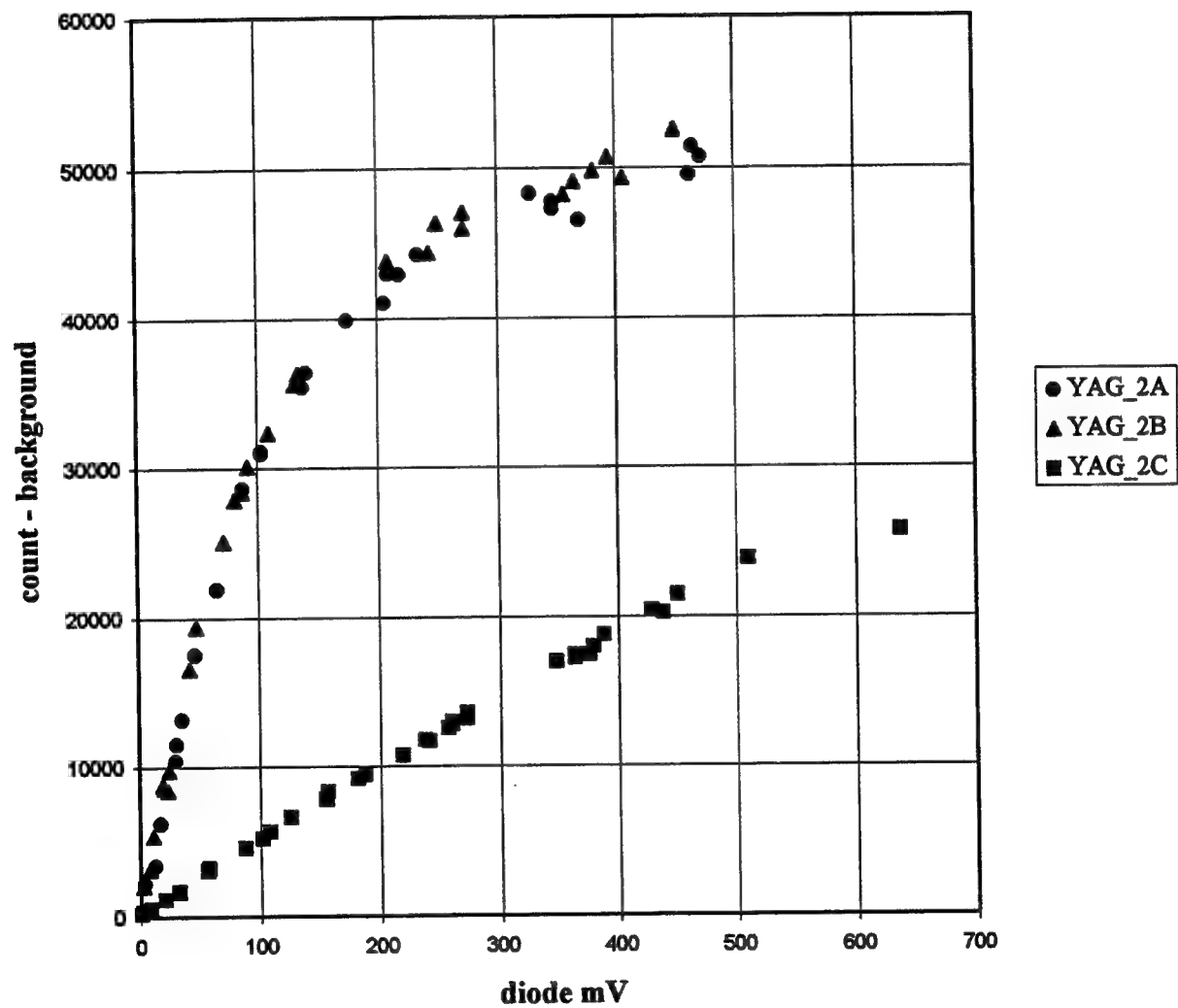


Figure 4

Camera 2 Linearity: YAG Source

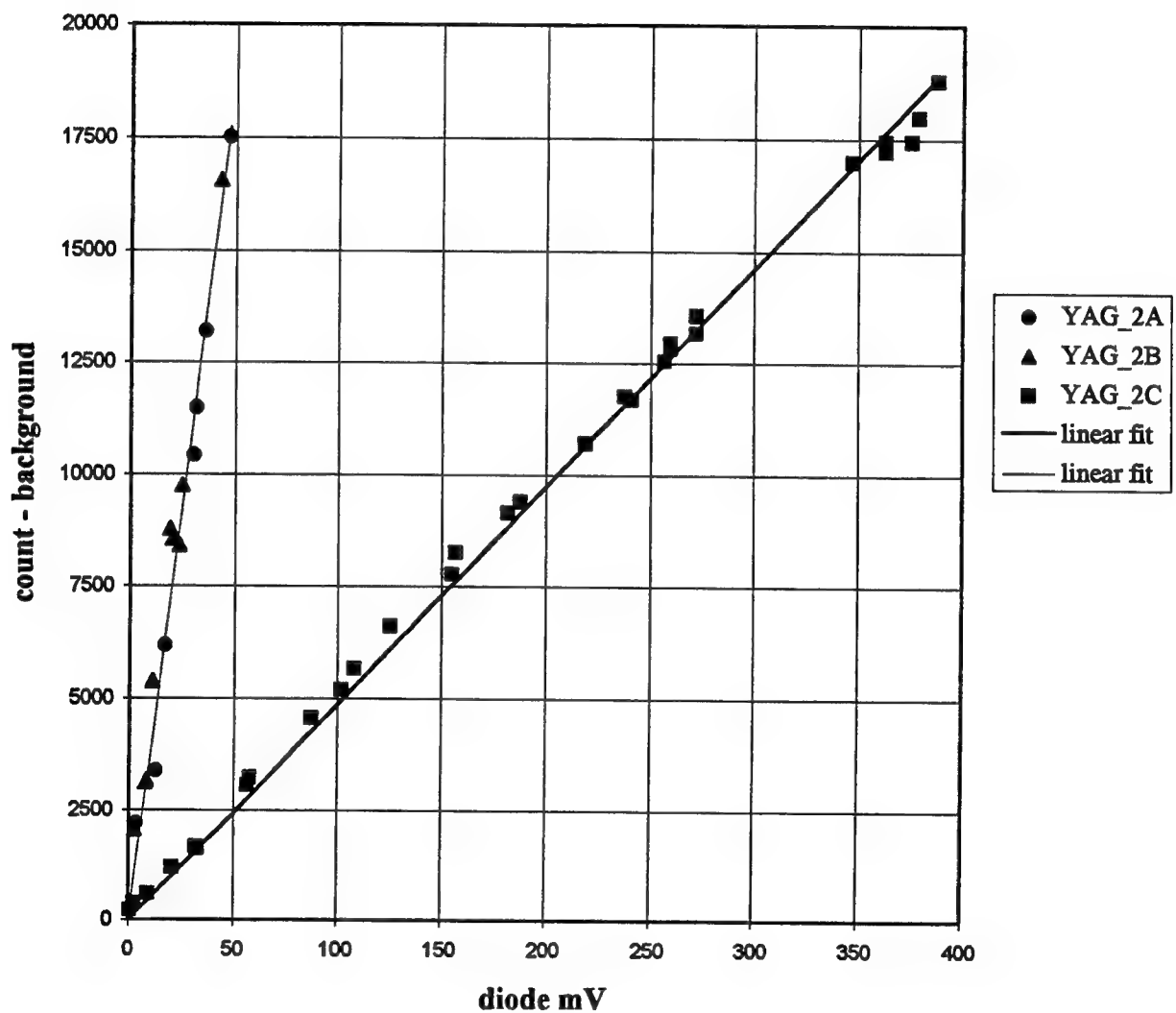


Figure 5

The Effect of Corrosion on Fatigue Crack Growth Rates in
Aircraft Structural Aluminum Alloys

James David Baldwin
Assistant Professor
School of Aerospace and Mechanical Engineering

University of Oklahoma
Norman, OK

Final Report for:
Summer Research Program
Oklahoma City Air Logistics Center

Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington DC

and

Oklahoma City Air Logistics Center

August 1995

THE EFFECT OF CORROSION ON FATIGUE CRACK GROWTH RATES IN AIRCRAFT STRUCTURAL ALUMINUM ALLOYS

J. David Baldwin
Assistant Professor
School of Aerospace and Mechanical Engineering
The University of Oklahoma

Abstract

One of the goals of the C/KC-135 Corrosion Fatigue program is to quantify the effect prior corrosion damage has on the fatigue strength of aluminum structural alloys. To support this goal, fatigue crack growth rates were recorded for artificially-corroded specimens of aluminum alloys 2024-T3, 2024-T4 and 7075-T6. The artificial corrosion was previously classified by Boeing as "severe." Corrosion reduced test specimen thickness approximately 3-5% as verified with spot measurements using a pointed tip micrometer. For a given alloy, specimens were tested in two environments (< 15% RH air and > 85% RH air) and at two stress ratios ($R = 0.05$ and $R = 0.50$). Two specimens per environment/stress ratio pair were tested, for a total of eight specimens per material. All of the tests were run at a constant amplitude cyclic load frequency of 10 Hertz.

Very rough estimates of noncorroded material crack growth behavior were made for comparison with the results from the corroded material. The general trend found was that the corroded material experienced higher crack growth rates than the noncorroded material. As expected, it was also observed that the more aggressive environment (> 85% RH air) and the higher stress ratio ($R = 0.50$) accelerated crack growth rates in the artificially-corroded materials.

THE EFFECT OF CORROSION ON FATIGUE CRACK GROWTH RATES IN AIRCRAFT STRUCTURAL ALUMINUM ALLOYS

J. David Baldwin

Introduction

The U.S. Air Force contingent of C/KC-135 aircraft is an aging fleet, the youngest having been delivered in 1965. Currently there is no plan to replace the fleet in the near future. As these aircraft age, they become more susceptible to corrosion damage, especially in the fuselage lap joints and wing skins around fastener holes. At this time, however, very little data exists on the reduction in strength of corrosion damaged materials. Because the effect of corrosion on structural strength has not been allowed for in the design of these aircraft, current inspection intervals may be too long to guarantee structural safety.

The goal of this program was to perform tests to quantify the fatigue crack growth rate in corroded aircraft aluminum. This basic material response data, when compared with data for noncorroded material, will provide an estimate of the magnitude of fatigue damage acceleration due to corrosion. Once a suitable body of data is available, it may then be integrated into existing structural integrity program procedures. Testing was carried out in the Engineering Testing Laboratory, Materials Test Facility, at Tinker Air Force Base, Oklahoma.

Discussion of Problem

The fleet of C/KC-135 aircraft spends most of its time on the ground exposed to a spectrum of atmospheric contaminants that promote corrosion of the structure. During the ground time, the airframe is essentially unloaded. On the other hand, fully loaded conditions occur during the relatively short periods when the aircraft are in flight (and in a relatively unaggressive environment at altitude). Because of this loading characteristic, it may be assumed that the loading cycles occur such that fatigue cracks will nucleate and grow, in a relatively unaggressive environment, in metal that has already experienced corrosion and a corrosive environment.

It is interesting to note that essentially all of the published work in corrosion fatigue has focused on the behavior of new (i.e., noncorroded) materials in various corrosive environments [1]. Although the existing literature provides valuable insight into the corrosion fatigue behavior of metals, it fails to address the metal's response to loading

after corrosion has already occurred. Only recently have experiments designed to quantify the fatigue response of corroded metal begun to appear [2,3]. The C/KC-135 Corrosion Fatigue test program was designed to collect some basic fatigue crack growth rate data for pre-corroded materials and compare it with data for baseline, noncorroded samples of the same materials.

The primary hypothesis of this investigation is that by examining in the laboratory the fatigue response of pre-corroded material in relatively unaggressive environments (specifically dry air and moist air), we can model the behavior of actual structural materials in flight. Note that this environmental assumption is probably not valid for aircraft spending their flight time in marine environments where a salt spray atmosphere may be more appropriate.

Methodology

The test program guiding this effort was defined in the C/KC-135 Corrosion Fatigue test protocol [4] developed by Boeing Defense & Space Group, Product Support Division (PSD) for the USAF. Carefully machined specimens were to be tested so that crack lengths (a) as a function of loading cycles (N) could be recorded. Post-test data reduction would convert the a versus N data into the standard fracture mechanics format of da/dN versus ΔK plots, where K is the crack tip stress intensity factor. All of the tests were to be conducted according to ASTM E-647 [5] as amended by the test protocol [4].

The total test matrix included as blocking variables aluminum alloy (2024-T3, 2024-T4, 7075-T6, 7178-T6), corrosion state (noncorroded, artificially corroded), environment (< 15% relative humidity air, > 85% relative humidity air), stress ratio ($R = 0.05$, $R = 0.50$) and loading frequency (0.1 Hz., 10 Hz.). The stress ratio, R , is defined as

$$R = \frac{\sigma_{\min}}{\sigma_{\max}}$$

where σ_{\min} and σ_{\max} are the minimum and maximum stress values in the loading waveform. The test program was divided among three laboratories; the material assigned to Tinker AFB was all artificially corroded, 10 Hz. specimens. A total of eight specimens were fabricated for each alloy, two for each humidity/stress ratio pair. Due to time constraints, the 7187-T6 specimens were not tested.

Load Frame: All of the testing was carried out in the Engineering Testing Laboratory, Materials Test Facility, at Tinker Air Force Base, Oklahoma using a servohydraulic material test system (MTS Systems Corporation, Minneapolis MN); the test system was calibrated in accordance with ASTM E-4 [6]. Testing was performed using a 20 kip load cell calibrated to a 2 kip range. Although it was not required by the ASTM E-647 test specification, the alignment of the load train was checked in accordance with ASTM E-1012 [7] before testing began. The average percent bending was observed to be less than 2%, indicating excellent alignment.

Specimen Preparation: The test specimens were prepared by Boeing PSD from material harvested from a government-provided retired C/KC-135 aircraft. When new, the 2024-T3, 2024-T4 and 7075-T6 materials were to have been clad. It was not known, however, how much of the original cladding was intact on the tested material. The 7178-T6 material was to have been bare when new. All of the specimens were machined according to ASTM E-647 in the Middle Tension configuration as shown in Figure 1. The 2024-T3, 2024-T4 and 7075-T6 specimens had a nominal width, W , of 1.75 inches and a nominal thickness of 0.060 inches; the machined starter notches, $2a_n$, were 0.350 inches long by 0.050 inches wide.

The specimens were corroded on one side using a protocol developed by Boeing PSD. In the test section, the noncorroded side of the specimen was polished to make the crack tips more visible. Each specimen was given a unique serial number for identification purposes and was accompanied by a blank test data sheet specifying the maximum far-field stress (remote from the crack tips), stress ratio, test frequency and test environment for that specimen. The far-field stress was chosen by Boeing to give initial crack growth rates of approximately 1.0×10^{-5} inches per cycle.

According to the test protocol, the corrosion state was to be "severe," but no further description was given. A rough estimate of the material loss in the specimens was made by measuring the thickness in the corroded area with a pointed tip micrometer. Random sampling of the specimen thicknesses showed that an average of less than 5 percent of the thickness had been lost to corrosion. It should be noted, however, that this method of quantifying corrosion is extremely approximate and was used only to make an order of magnitude estimate of the material loss. Other research groups are investigating more rigorous methods for quantifying corrosion damage that will be useful in future material characterization studies.

Environmental Controls: A two-piece, clear plastic environmental chamber was made to enclose the test

section of the specimen; each of the identical halves of the test chamber had a sliding window giving access to the test section. When combined, the test chamber was approximately 4 inches wide by 2 inches tall by 2 inches deep; the chamber halves were closed around the specimen and sealed by strips of M-Coat FB Butyl Rubber Sealant (Micro-Measurements Division, Raleigh NC).

Two atmospheric conditions were maintained in the laboratory. A dry air stream at < 15% relative humidity was created by passing an air stream through a cylinder packed with desiccant. Two aquarium air pumps supplied the air through plastic hoses; another pair of hoses connected the specimen environmental chamber to the desiccant cylinder and an outlet vacuum pump, respectively. The air stream relative humidity was monitored at the vacuum pump inlet. A moist air stream (>85% RH) was created similarly, except the air pumps bubbled air up through a cylinder of Type III reagent water per ASTM D-1193 [8]. Again, the relative humidity was monitored at the outlet of the environmental chamber. The relative humidity was controlled by adjusting the input voltage to an aquarium heater immersed in the water cylinder (i.e., increasing the water temperature increased the output relative humidity).

Crack Length Measurement: A Gaertner traveling microscope (with approximately 32X magnification) and a Gaertner micrometer slide (with 0.0001 inch resolution), were used to observe and measure all crack lengths (Gaertner Scientific Corp., Chicago, IL). Because the corrosion would obscure the crack tip locations, the crack was observed only on the polished side of the specimen. The tip-to-tip crack length, $2a$ in Figure 1, was measured.

ASTM E-647 guidelines for growing a crack from the machined notch of a test specimen were observed. The pre-cracking rules specify the minimum length at which valid crack length measurements can be obtained during a test. Following ASTM E-647, the specimen was cycled at the prescribed load until a crack length, a , of 0.050 inches was observed. To reduce the time required to pre-crack a specimen, it was cycled at 20 Hz.; after the pre-crack threshold was reached, the cycling frequency was reduced to the specified 10 Hz.

Data Reduction and Analysis: The crack length versus cycle count data were converted to da/dN versus ΔK using a modified version of the Fortran program listed in ASTM E-647. A seven point incremental polynomial curve fit was used to estimate the derivative da/dN and the stress intensity range was computed by [5]

$$\Delta K = \frac{\Delta P}{B} \sqrt{\frac{\pi a}{2W} \sec \frac{\pi a}{2}}$$

where ΔP is the cyclic load range $P_{\max} - P_{\min}$, B is the specimen thickness and $\alpha = 2a/W$.

Results

Because the raw test data for noncorroded specimens in this program were unavailable for comparison, the corroded specimen data will be compared with rough estimates of noncorroded material behavior. Mr. Joe Luzar of Boeing PSD supplied some preliminary da/dN versus ΔK plots for the noncorroded materials tested at other laboratories. From these plots, a "visual best fit" straight line was drawn through the data and the endpoints of the line were estimated from the graph. The noncorroded material behavior is represented in the following graphs using this estimate.

Figures 2 and 3 show the crack growth data for corroded 2024-T3 alloy; the open data points are labeled "BL" for baseline, or noncorroded, material. For the $< 15\%$ RH tests, two baseline data lines are shown for each stress ratio because of scatter in the Boeing data. Note that two corroded specimens were tested for each environment/stress ratio combination, except the $< 15\%$ RH, $R=0.05$ case where only one specimen was tested.

In each Figure, the data exhibit linear (i.e., power law, $da/dN \sim \Delta K^m$) behavior with some notable exceptions. One of the $< 15\%$ RH, $R=0.50$ and one of the $> 85\%$ RH, $R=0.05$ specimens each show significant departures from linearity. It is felt that these curving data plots are the result of difficulty finding the crack tip at short lengths. As the crack grew in length, it generally became easier to see through the microscope.

The most notable result to be seen in these data is that the corroded material appears to have crack growth rates exceeding the corresponding baseline, noncorroded material. The logarithmic axes tend to downplay the magnitude of the difference, but a close examination shows corroded crack growth rates increased by as much as a factor of two over the baseline data. An exception to this trend is the $< 15\%$ RH, $R=0.05$ data, where the corroded data lies between the two baseline lines. It should be remembered, however, that the corroded data represents only two specimens, that the baseline data has been estimated from graphs, not from data points, and that there will be scatter in both the corroded and baseline data that have not been quantified here.

It can be seen from these data that the corroded material also exhibits the anticipated behavior with respect to stress ratio and environment. First, if the two figures were laid on top of each other, the data for $> 85\%$ RH would be typically above and to the left of the $< 15\%$ RH specimens. It has been generally observed that more aggressive

environments lead to faster crack growth and this data reflects that trend. Also, it can be seen that the $R=0.50$ data generally lies above and to the left of the $R=0.05$ data. Recalling that increasing stress ratios imply larger mean stresses (which are known to accelerate fatigue failure), we would expect such a result.

Figures 4-5 and 6-7 show the crack growth rate data for 2024-T4 and 7075-T6 alloys, respectively. Again, there were two corroded specimens per data series. In these cases, we again see the previously observed (and expected) behavior with respect to corrosion, environment, stress ratio and power law behavior. Exceptions are seen in Figure 4, where the 2024-T4, $< 15\%$ RH, $R=0.50$ data lies within the baseline band, in Figure 6, where the 7075-T6, $< 15\%$ RH, $R=0.05$ data shows an unexplained "hump" and in Figure 7, where the 7075-T6, $> 85\%$ RH, $R=0.50$ data seems to have no linear region. However, it is important to note that despite these experimental oddities, the corroded material consistently appears to grow cracks at a higher rate than the baseline material.

Conclusions

The goal of this investigation was to acquire fatigue crack growth rate data for artificially-corroded aircraft structural aluminum alloys. The data showed that the corroded material appears to experience higher crack growth rates than noncorroded material. It should be remembered, however, that the noncorroded material response used in this comparison was based on visual curve fits from baseline material test data. As expected, the corroded materials showed increased crack growth rates at higher stress ratios and in more aggressive environments.

The issues of data scatter and statistical significance have not been addressed in this study. It is unclear at this time whether the corroded material crack growth rates can be shown to be higher than those for noncorroded material based on statistical hypothesis testing. Such a demonstration will be essential before any conclusions can be drawn about the effects of corrosion on structural integrity.

While the results of this program provide some tantalizing hints about corrosion's impact on material performance, they are preliminary and need to be augmented by further testing. The most significant factor that has yet to be included in the test matrix is a quantifiable corrosion severity. Here, an approximate (and by no means rigorous) measure of the material loss was made by sampling the thickness of the corroded material using a pointed tip micrometer. Currently, there are no objective measures of corrosion damage, but other researchers are at work creating

such a classification system. Future testing conducted to explore the anticipated crack growth acceleration as a function of the degree of corrosion will then be able to use corrosion severity as an experimental variable. Such data will be necessary for future efforts at incorporating the potential for corrosion damage into structural integrity calculations.

References

- [1] Gangloff, R.P., *Corrosion Fatigue Crack Propagation in Metals*, NASA CR 4301, Washington, D.C.: National Aeronautics and Space Administration, 1990.
- [2] Scheuring, J.N. and A.F. Grandt, "An Evaluation of Fatigue Properties of Aging Aircraft Materials," presented at the 1994 USAF Structural Integrity Program Conference, San Antonio TX.
- [3] Koch, G. H., "The Effects of Pre-Existing Corrosion on the Fatigue Cracking Behavior of Aircraft Aluminum Alloys", to be presented at the USAF Third Aging Aircraft Conference, Wright-Patterson AFB OH, 1995.
- [4] *Integrated C/KC-135 Corrosion Program Round Robin Testing: Pre-Corroded Fatigue Crack Growth Rate Test Plan*, Wichita KS: Boeing Defense & Space Group, Product Support Division, March 1995.
- [5] ASTM E-647, *Test Method for Measurement of Fatigue Crack Growth Rates*, Philadelphia PA: American Society for Testing and Materials, 1993.
- [6] ASTM E-4, *Standard Practices for Force Verification of Testing Machines*, Philadelphia PA: American Society for Testing and Materials, 1994.
- [7] ASTM E-1012, *Standard Practice for Verification of Specimen Alignment Under Tensile Loading*, Philadelphia PA: American Society for Testing and Materials, 1993.
- [8] ASTM E-1193, *Standard Specification for Reagent Water*, Philadelphia PA: American Society for Testing and Materials, 1983.

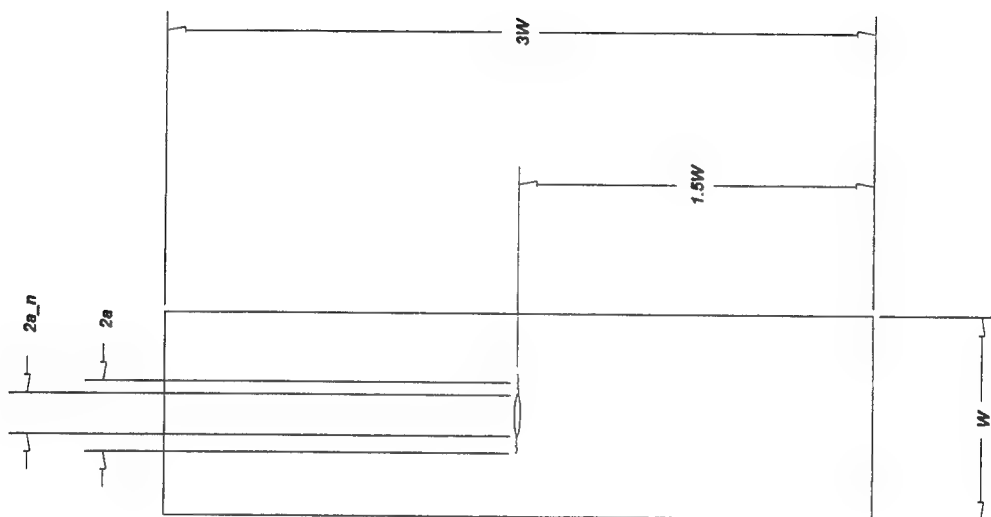


Figure 1: Schematic Definition of the ASTM E-647 Middle Tension Specimen [5]

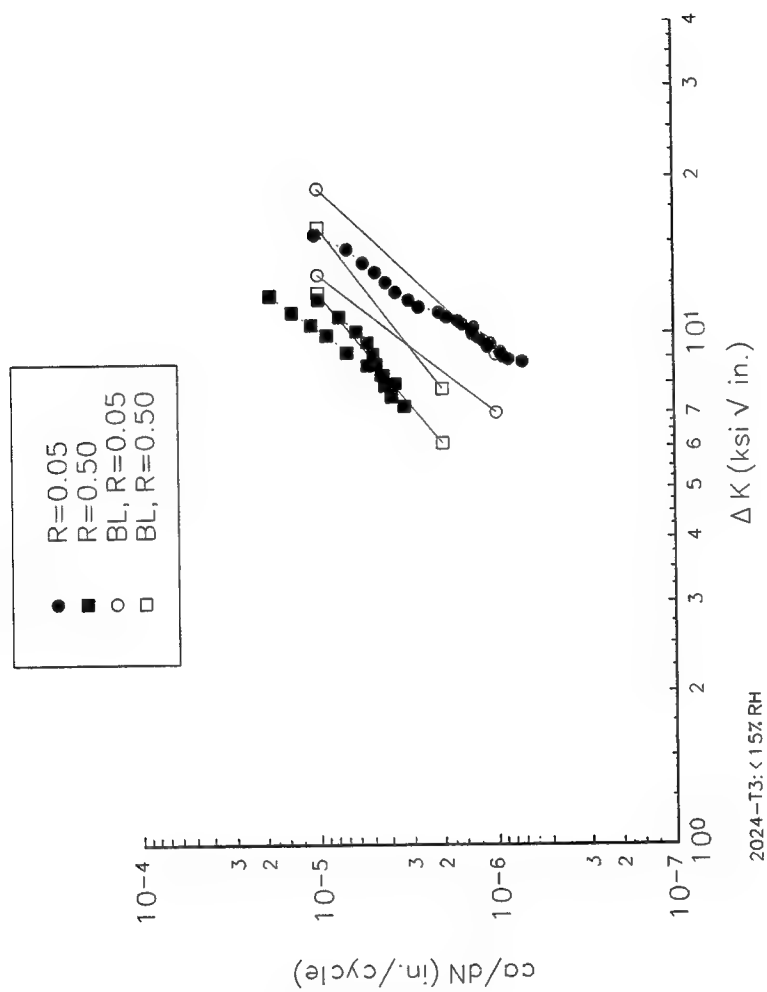


Figure 2: Fatigue Crack Growth Rates, Alloy 2024-T3, < 15% Relative Humidity

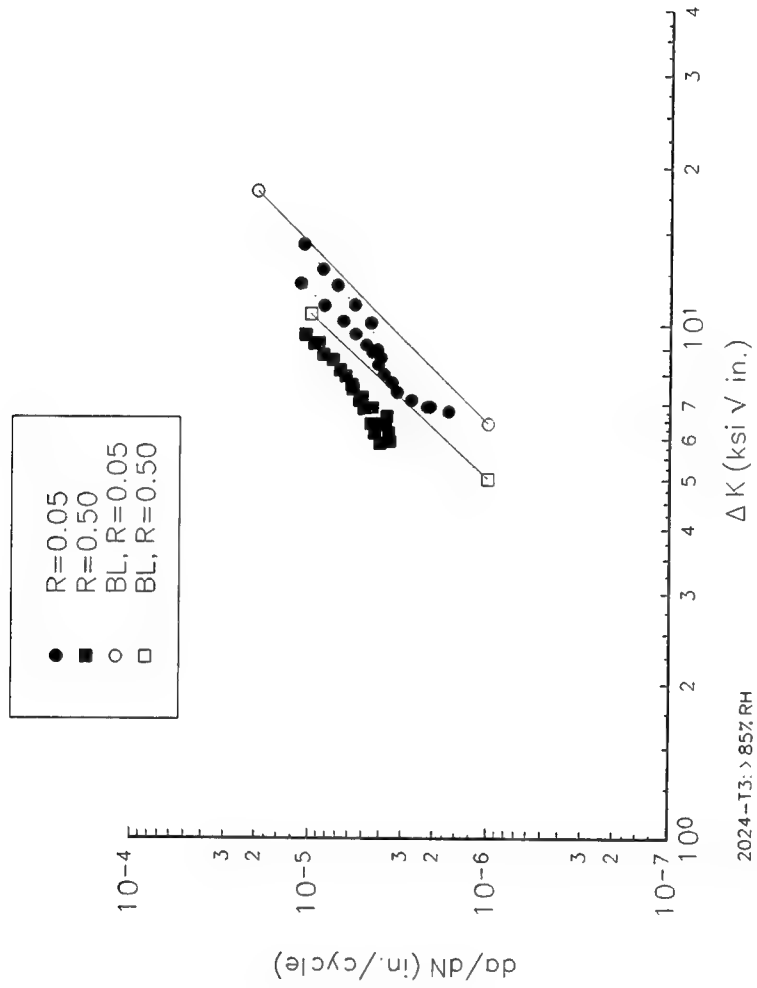


Figure 3: Fatigue Crack Growth Rates, Alloy 2024-T3, > 85% Relative Humidity

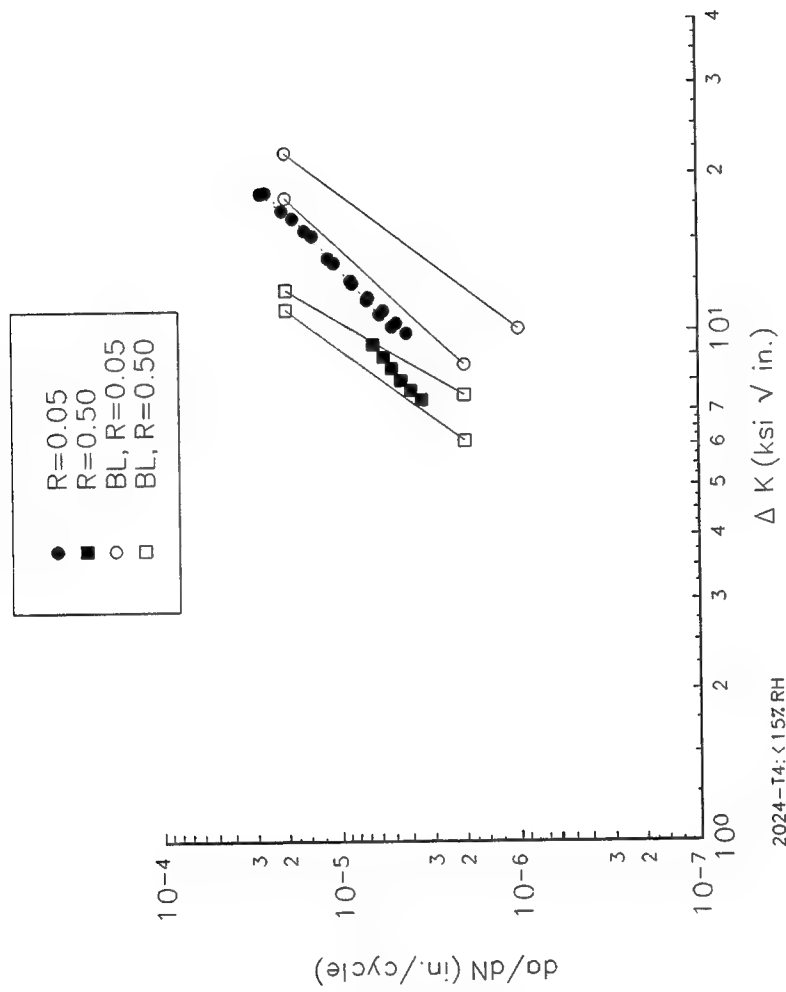
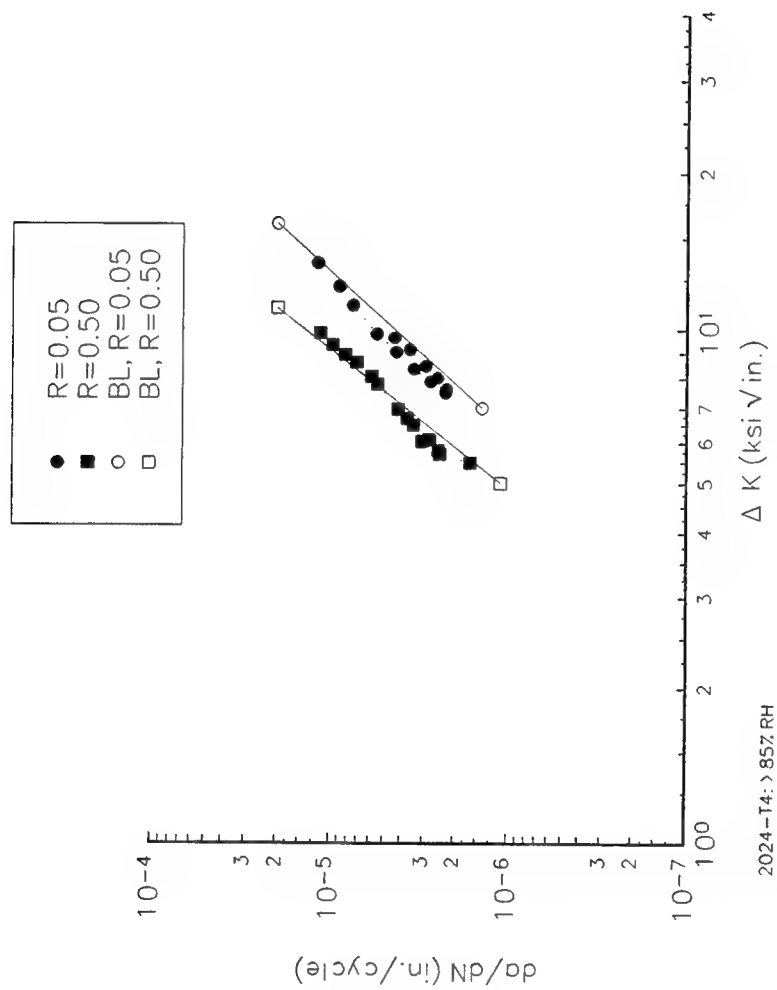


Figure 4: Fatigue Crack Growth Rates, Alloy 2024-T4, < 15% Relative Humidity



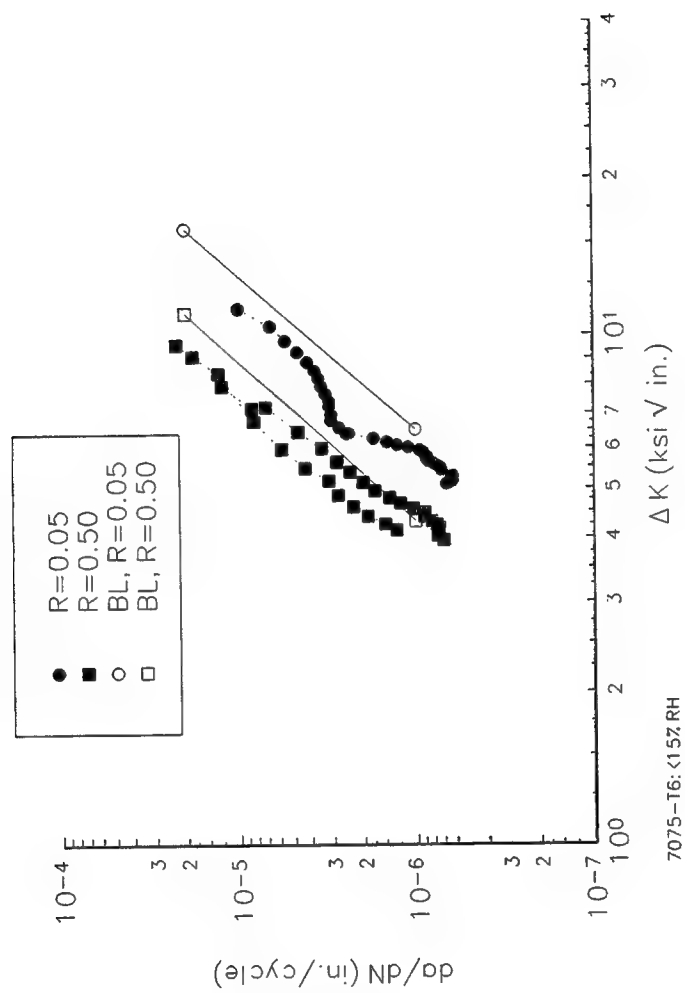


Figure 6: Fatigue Crack Growth Rates, Alloy 7075-T6, < 15% Relative Humidity

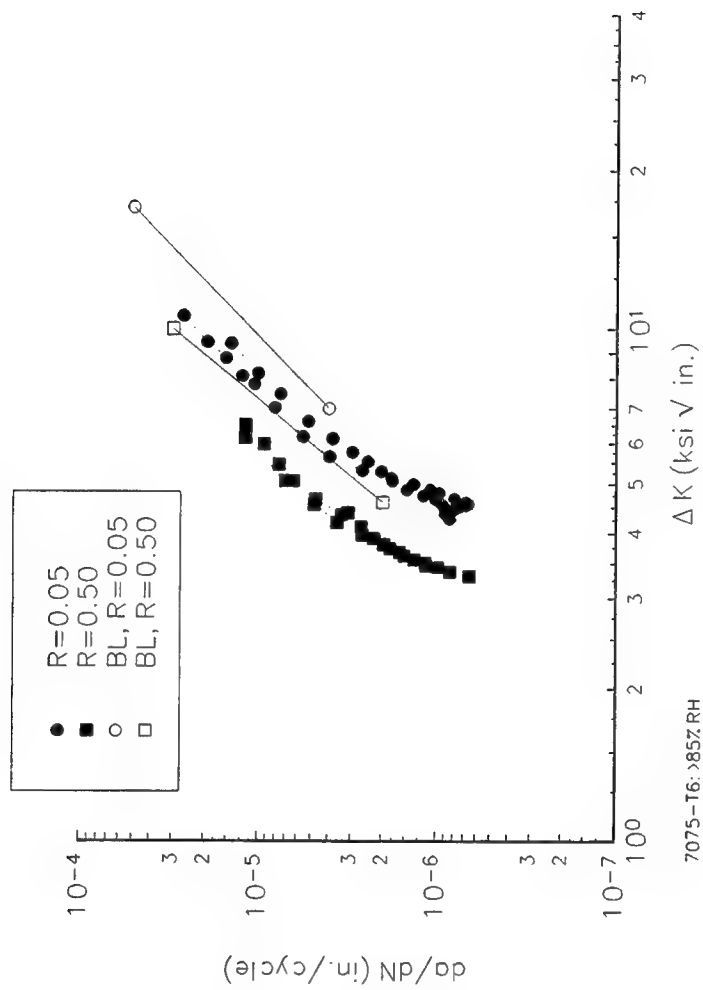


Figure 7: Fatigue Crack Growth Rates, Alloy 7075-T6, > 85% Relative Humidity

**An Alternative to Silane Coupling Agents as Adhesion Promoters:
Admicellar Polymerization of Styrene-Isoprene
on Glass Cloth for Use in Composite Manufacture**

**Brian P. Grady
Assistant Professor
Department of Chemical Engineering**

**University of Oklahoma
Norman, OK**

**Final Report for:
Summer Research Program
Oklahoma City Air Logistics Center**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington DC**

and

Oklahoma City Air Logistics Center

August 1995

AN ALTERNATIVE TO SILANE COUPLING AGENTS AS ADHESION PROMOTERS:
ADMICELLAR POLYMERIZATION OF STYRENE-ISOPRENE ON GLASS CLOTH FOR
USE IN COMPOSITE MANUFACTURE

Brian P. Grady
Assistant Professor
Department of Chemical Engineering
University of Oklahoma

Abstract

A new method for promoting the adhesion of thermosetting resins to a reinforcing agent was successfully demonstrated. In this method, termed admicellar polymerization, monomers are polymerized inside a surfactant bilayer adsorbed on a surface. Admicellar polymerization is essentially the surface analogue of emulsion polymerization. This method has significant advantages over current silane coupling technology, primarily admicellar polymerization is less expensive and more robust.

In this study, glass mats were coated with styrene-isoprene copolymers using admicellar technology and the resulting composites made with polyester or epoxy resin were compared to the silane and untreated counterparts. *For both types of resins, the composite made from the admicellar treated material had a higher average ultimate strength than the composite made from the untreated material.* In the epoxy material, the improvement was close to that provided by the coupling agent. Since the admicellar polymerization process was not optimized, the significant conclusion of this study is that admicellar polymerization improves the adhesion between the resin and the reinforcing filler in a thermosetting composite; however the maximum amount of improvement has not yet been determined.

AN ALTERNATIVE TO SILANE COUPLING AGENTS AS ADHESION PROMOTERS: ADMICELLAR POLYMERIZATION OF STYRENE-ISOPRENE ON GLASS CLOTH FOR USE IN COMPOSITE MANUFACTURE

Brian P. Grady

Introduction

Silane coupling agents improve the adhesion of inorganic materials to polymers and enjoy widespread commercial application, including uses in motor-vehicle tires¹ and aircraft radomes. The disadvantages of using silane-coupling agents as adhesion promoters is well-known, in particular these agents are very expensive, water sensitive, only useful for surfaces containing certain chemical groups, and different agents must be used for different applications. The purpose of the research described in this proposal was to evaluate admicellar polymerization as an alternative to silane coupling agents as adhesion promoters for thermosetting composites.

The Basics of Admicellar Polymerization

Admicellar polymerization is a term used to describe a novel method of synthesizing polymers on the surface of a solid substrate. This polymerization occurs in the hydrophobic region of a surfactant bilayer adsorbed on the surface. The commercial potential of admicellar polymerization is quite large for the following three reasons: 1) the process is extremely robust; almost any surface can be coated with almost any polymer, 2) overall process cost is low, and 3) adhesion properties of a solid depend almost exclusively on the surface character.

Admicellar polymerization was developed from a fundamental understanding of the interaction between surfactants and surfaces. It was known that the conditions of a water solution containing surfactant and a suspended inorganic particle could be tailored to favor the formation of a bilayer of surfactant on the surface of the particle rather than either a monolayer of surfactant, a normal micelle or isolated surfactant molecules.² Other experimental^{3,4} and theoretical evidence⁵ has proved this postulate. This mechanism of surfactant adsorption on a surface necessarily causes a hydrophobic region in the interior of the bilayer as illustrated in Figure 1. Later, it was hypothesized that the environment in the hydrophobic region of the bilayer might serve as the medium for a polymerization, as the hydrophobic region inside spherical micelles serves as the medium in an emulsion polymerization.

This hypothesis has proven to be correct with the admicelle functioning as an interfacial polymerization solvent. The system styrene-sodium dodecyl sulfate (SDS)-alumina was the first system studied⁶ and formed the basis of the first patent in this area.⁷ Two other patents on admicellar polymerization have followed.^{8,9} Admicellar polymerization has proven to be a very versatile process. Different monomers which have already been admicellar polymerized include tetrafluoroethylene,¹⁰ pyrrole¹¹, isoprene, butadiene and styrene-diene copolymers,¹² while titanium dioxide¹³ and silica¹² have also proven to be suitable substrates for this polymerization. Selection of the proper surfactant is a key element to the successful manipulation of these systems, and cationic, anionic and nonionic surfactants have been used. Polymerizable surfactants have also been shown to participate in the polymerization.^{14,15}

Figure 1 illustrates the four-step admicellar polymerization process. Step 1 consists of admicelle formation through the adsorption of a surfactant bilayer onto the surface of the

substrate. Adsorption is accomplished by the use of a suitable surfactant under appropriate system conditions. The choice of surfactant is influenced by the point of zero charge (PZC) of the substrate, the chemical nature of the polymer to be formed, and the choice of initiator. A study of the PZC for the substrate provides information on the pH ranges in which cationic or anionic surfactants may be utilized. The instability or dissolution of components in certain pH ranges may further define the range of useful operating conditions, and hence, surfactant type. The quantification of the amount of surfactant required is achieved through the determination of adsorption isotherms, which is the measurement at a given temperature of the amount of surfactant adsorbed on the surface as a function of the solution surfactant concentration.

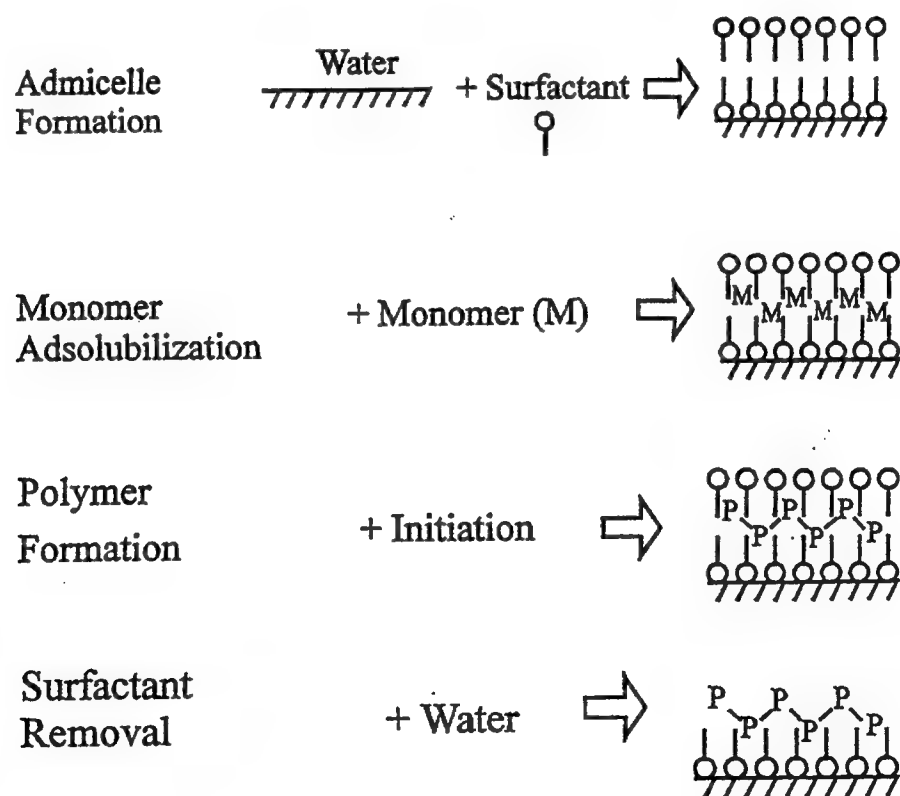


Figure 1. The Admicellar Process

Step 2 in the process is the solubilization of monomer into the admicelle, a phenomenon termed adsolubilization. Many organic monomers are nearly insoluble in water. Thus, at equilibrium, they preferentially partition into the hydrophobic interior of the admicelle. This process can occur after the formation of the admicelles, or concurrently with surfactant adsorption. It is convenient experimentally to dissolve the monomer, and often the initiator, in the surfactant feed solution prior to surfactant adsorption. The presence of micelles promotes the solubility of the monomer in the feed solution, increases the rate of adsolubilization of the monomer in the admicelles, and helps prevent the formation of emulsions. When the feed solution is contacted with the substrate, the adsorption of the micelles is thought to carry the solubilized monomer onto the substrate surface.

Step 3 is the in-situ polymerization of the monomer. For free radical polymerization, the reaction is started by the generation of radicals from the initiator. Once the reaction has begun, additional monomer from the bulk solution diffuses into the admicelle.¹⁶ If the reaction is continued for a sufficient length of time, essentially all of the monomer dissolved in the solvent can be converted to polymer. Step 4 is the washing of the treated substrate to remove the outer surfactant layer to expose the polymer film.

The result of an admicellar polymerization is a surface covered with a very thin polymer layer. Because surfactant adsorption is almost universal, this method offers a mechanism to coat arbitrary surfaces with a polymer. For reasons not entirely understood, the polymer seems to be strongly attached to the surface. If the proper monomer is used, the admicellar polymer has the capability of reacting with a bulk polymer which can lead to a significant improvement in the adhesion properties of the polymer to an inorganic substrate. Our most extensive

investigations have been with precipitated silica in rubbers, where it has been proven that admicellar polymerization on the surface of the silica improves the adhesion of the rubber to the silica.¹² In this study, styrene-isoprene copolymers were polymerized on the surface of glass cloth, and the mechanical properties of the composite made from this glass cloth were compared to composites manufactured using the untreated and silane treated glass cloths respectively.

Methodology

Materials

Styrene monomer was purchased from Scientific Polymer Products while isoprene monomer was obtained from Aldrich Chemical. The monomers were stored at -40°C prior to use. Immediately prior to use, the styrene inhibitor was removed by passing the monomer through a column supplied by the vendor. The initiator, VA-044, was supplied by Wako Chemical and also stored in the freezer. The surfactants, cetyl trimethyl ammonium bromide (CTAB) and sodium dodecyl sulfate (SDS) were purchased from Sigma Chemical. Series 181 cloth, before and after reaction with a silane coupling agent, was supplied by the Clark-Schwebel company. High pressure liquid chromatography (HPLC) solvents were available in the Composites laboratory. Except as mentioned above, all materials were used as received. All water used in these experiments, except that required for HPLC, was distilled through a Wheaton Lab Still and stored in Nalgene containers.

Adsorption Isotherms

Surfactant concentration standards were used to prepare a calibration line relating HPLC

integrated area to surfactant concentration. To determine the amount of surfactant adsorbed on the glass, the mats were cut into squares and immersed in 50.0 ml of water containing a measured amount of the surfactant for at least two days. The integrated area was measured before and after immersion and the amount of surfactant adsorbed on the cloth was calculated using the difference in concentration before and after immersion. The water was made basic using potassium hydroxide or acidic using sulfuric acid.

A 100 x 4.6 mm Hypersil ODS 5 micron column from Hewlett Packard was used for SDS analysis, while a 50 mm x 4.6 mm Cation R column from Alltech Associates was used for the CTAB analysis. For the former a 60/40 MeOH/water at 1.00 ml/min mix was used as a carrier solvent, while 100% water at 0.500 ml/min was used as a carrier solvent for the latter. The HPLC was a Hewlett-Packard Series 9000 equipped with a Pascal Chemstation and the detector an Alltech 350 Conductivity detector interfaced to a HP 35900 analog-to-digital convertor. The column temperature was maintained at 40°C, while the conductivity measurements were made at room temperature.

Admicellar Polymerization

The polymerization was performed in a 5 gallon Nalgene carboy. The glass cloth was added to the carboy containing approximately 3.5 gallons of water; because of its density the cloth sunk to the bottom. The surfactant was added and the solution was maintained at room temperature for 6 hours with periodic agitation. The monomers and initiator were added and the carboy was transferred to an oven at 60°C for 60 hours. The mole ratio of styrene:isoprene was 1:1, while the overall surfactant to monomer ratio was also 1:1. If

the whole surface of the cloth were covered uniformly with polymer and all the monomer reacted to form polymer, the thickness of the film should have been about two monomer units based on the surface area of the cloth calculated from the adsorption isotherms. The surfactant to initiator mole ratio was 10:1, which is quite high compared to emulsion polymerization but low compared to admicellar polymerization on silica.

After 60 hours at 60°C, the carboy was removed from the oven, vented under a hood, and the glass cloth removed. The glass cloth was washed in DI water and dried at room temperature. Parts of the cloth had become a very light yellow due to the polymerization. The color was definitely not uniform, suggesting that the thickness of the polymer film was also not uniform. This non-uniformity is undoubtedly a result of the polymerization conditions, in particular the lack of agitation.

Wet Layup and Mechanical Testing

Two types of resins were used in these experiments, an epoxy resin and a polyester resin. EPON® 828 epoxy resin from E.V. Roberts Inc. cured with the ethyleneamine curing agent DEH 26 from Dow Chemical was used for the former and a low pressure polyester laminating resin from Chemical Commodities cured with methyl ethyl ketone peroxide from Ciba-Geigy was used for the latter. All chemicals were used as received. The chemicals were thoroughly mixed at the recommended weight ratio and then liberally applied to each ply. A total of three plies were used. The material was vacuum bagged with bleeder and breather for four hours at room temperature, then the bagging material was removed and the product was cured at 120°F overnight.

Three point bend testing was performed on a SFM-110 tensile tester manufactured by United Calibration Corporation. A 2" x 0.5" beam was used with a 1" span, according to the specifications described in ASTM D790. A crosshead speed of 0.05 in/min was used, with a preload of 0.2 lbs. In no cases did the experimental strain level reach the maximum recommended value as detailed in the ASTM test method. The modulus and maximum stress on the outer fiber were calculated according to the formulas given in the ASTM standard. The results from five bars, each specimen being cut from a different part of the sample, were averaged for each layup result.

Results and Discussion

Adsorption Isotherms

Since both glass cloth and silica are based on SiO_2 , it was thought that the same surfactant and pH could be used. For silica, the appropriate surfactant is CTAB which will adsorb at pH=8. Unfortunately, it certainly did not appear that CTAB adsorbed to the glass cloth at pH=8 or under even more basic conditions. The reason a stronger conclusion cannot be made is that the CTAB eluted at the same time as the injection peak and it was difficult to determine if CTAB adsorbed in small amounts. Also, the height of the injection peak varied before and after glass cloth was added, suggesting that other chemical species were eluting with the CTAB.

To illustrate the problems with determining the concentration of CTAB via HPLC, at pH=8 the injection peak area with no CTAB was approximately 1.5. When this value was subtracted from the integrated area of the concentration standards, an excellent calibration line was generated covering the concentration range 0-10,000 micromolar. When glass cloth was added,

the height of the injection peak seemed to be 6 for the same set of HPLC conditions. This value was determined from the integrated area at very low CTAB concentrations; the HPLC results of the supernatant liquid were all identical and hence the assumption was made that all the CTAB had adsorbed. Subtracting 6 from the integrated peak areas for the higher concentration solutions contacted with the glass cloth resulted in a non-sensical isotherm. Overall, at low concentrations there was a difference in the HPLC integrated areas before and after contact with glass cloth which was attributed to a combination of zero shift in the injection peak and change in CTAB concentration, while at high concentrations these differences disappeared. The latter result, coupled with the lack of a reasonable isotherm, suggest that the CTAB did not adsorb on the glass cloth.

Under normal circumstances, the species to be analyzed is separated from the injection peak by using the appropriate column/carrier solvent combination. To try and separate the CTAB from the injection peak, three different C18 columns from two different manufacturers and the Cation R column were employed. On the advice of a colleague,¹⁷ various water/MeOH ratios were used with the former column in order to induce some holdup, however nothing in the extensive series of experiments suggested that any of these columns heldup the CTAB. The manufacturer of the Cation R column suggested that dilute nitric acid solutions should separate the CTAB, however various molarities from about 10 micromolar to 0 micromolar were tried to no avail. Higher acid concentrations were not tested for fear the HPLC might be damaged. Finally, the water was made acidic with sulfuric acid to see if any adsorption would occur at low pH and similar problems were found. Nothing that was tried suggested that the CTAB could be separated from the injection peak.

Fortunately, SDS did adsorb to the glass cloth in acidic conditions. The ODS column was able to separate the SDS from the injection peak so very accurate measurements were possible. Figure 2 is a plot of the adsorption isotherm for SDS on the glass cloth with no surface treatment and has the expected characteristic shape. For more information on adsorption isotherms, the reader is directed to the book chapter written by O'Haver *et.al.*¹⁸ Assuming a surface coverage of 35 \AA^2 per two surfactant molecules, the surface area/weight ratio of the glass cloth is approximately $3 \text{ m}^2/\text{gram}$. To prevent emulsion polymerization and maximize the amount of admicellar polymer, the amount of surfactant should be adjusted so that the operating point is right below the upper plateau. Hence, the desired surfactant concentration in solution was 6000 micromolar; the actual amount was measured via HPLC to be 6400 micromolar.

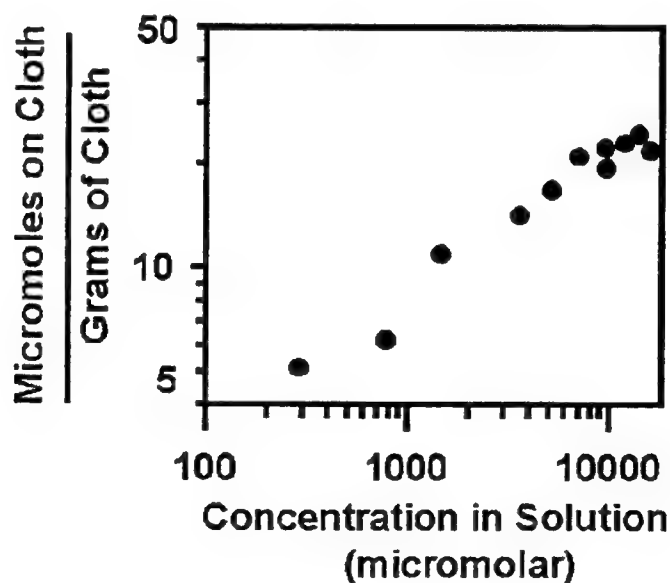


Figure 2: Adsorption isotherm for SDS on silica at $\text{pH} \approx 4$.

Lay-ups

The wetting behavior of the epoxy resin was qualitatively quite different for the silanated material since the epoxy resin was easier to apply to the silanated cloth either the untreated or admicellar treated material. Three-point bend test results are shown in Table 1 below with the errors representing one standard deviation. The moduli are not shown because the differences between the three composites were not statistically significant. The most sensitive measure of the adhesion between the resin and the glass cloth was the maximum stress value. The factor of two difference between the untreated and silane treated materials is consistent with differences reported in the literature.¹⁹ The important conclusion to draw from Table 1 is that the admicellar coating significantly improves the adhesion between the glass cloth and the resin.

TABLE 1

Results of Epoxy Resin Lay-ups

	Percent Elongation	Ultimate Strength (psi)
Silane	17.2 ± 2.6	45,000 ± 18,600
Untreated	10.0 ± 3.9	24,800 ± 7,400
Admicelle	12.6 ± 4.8	37,600 ± 4,700

To quantify the probability that the admicellar film does improve polymer/glass adhesion, the student t-test was employed. This procedure tests statistically whether the means of two sets of data are different. According to this test, there is a greater than 97.4% probability that the average ultimate strength of the silane treated material is different than the untreated material.

There is a greater than 91% probability that the results using the admicellar coated material are different than the untreated material. Finally, there is only a 64% probability that the ultimate strength of the silane treated material is different than the admicellar treated material.

Unfortunately, the standard deviations shown in Table 1 are quite large, in particular for the silane coupled material. The inherent error of the test is probably represented by the error for the untreated material, since its surface characteristics should be uniform. Even this error seems high; it is believed that the limited number of plies was the primary cause. The much higher error for the silane coupled material is probably due to poor uniformity of silane coupling agent leading to a very non-uniform surface. There is no other evidence to support this assumption, rather it seems the only logical explanation. Even though the coating was clearly non-uniform on the admicellar material, the variation in the physical properties was actually quite small. This observation probably indicates that only a very thin layer is required to reinforce the material since only those chains strongly anchored to the glass cloth can act as a reinforcing agent, i.e. chains increasing the thickness of the polymer film in a particular area do not add any benefit to polymer/glass cloth adhesion. Hence, the admicellar process may have the added benefit of producing a material with more uniform mechanical properties.

Table II shows the results for the polyester composite. In this case, the relative error for the three materials are roughly identical. The admicellar coating may not have significantly improved the flexural strength relative to the untreated material since the t-test indicated that only a 70% chance existed that the means of the two data sets were different. It should be noted that the silane treated material, unlike the admicellar and the untreated material, was not from the same base stock of glass cloth. Hence, perhaps the surface characteristics of this roll of

material was more uniform than the roll used for the epoxy composite studies.

TABLE 2

Results of Polyester Resin Lay-ups

	Percent Elongation	Ultimate Strength (psi)
Silane	11.7 \pm 3.1	26,800 \pm 3,600
Untreated	12.0 \pm 0.9	13,500 \pm 2,000
Admicelle	10.0 \pm 1.2	14,700 \pm 1,000

Conclusions

A new process for coating reinforcing fibers in thermosetting polymer composites to improve polymer/filler adhesion was demonstrated. This process, termed admicellar polymerization, uses soap, initiator and polymer monomer to coat a surface with a very thin layer of polymer. Glass mats were coated with styrene/isoprene copolymers in this manner and the coated material was used to make three ply glass/epoxy and glass/polyester composites. The composites made from the admicellar coated material had a higher flexural strength in three point bend tests than the composite made from the uncoated material. This study demonstrates the feasibility of improving polymer-filler adhesion in thermoplastic composites using admicellar polymerization; further studies are needed to optimize the monomer type, level and processing conditions to determine the maximum level of reinforcement available.

References

1. see for example Science and Technology of Rubber (2nd Edition), J.E. Mark, B. Erman and F.R. Eirich, Eds., (San Diego: Academic Press) 1994.
2. J.H. Harwell, J.C. Hoskins, R.S. Schechter and W.H. Wade, *Langmuir*, **1**, 251 (1985).
3. D. Bitting and J.H. Harwell, *Langmuir*, **3**, 500 (1987).
4. D.C. McDermott, J.R. Lu, E.M. Lee, R.K. Thomas and A.R. Rennie, *Langmuir*, **8**, 1204 (1992).
5. M.A. Yeskie and J.H. Harwell, *J. Phys. Chem.*, **92**, 2346 (1988).
6. J. Wu, J.H. Harwell and E.A. O'Rear, *Langmuir*, **3**, 531 (1987).
7. J.H. Harwell and E.A. O'Rear, U.S. Patent 4,770,906 (1988).
8. J.H. Harwell and E.A. O'Rear, U.S. Patent 4,900,627 (1990).
9. J.H. Harwell and E.A. O'Rear, U.S. Patent 5,106,691 (1992).
10. C.L. Lai, J.H. Harwell, E.A. O'Rear, S. Komatsuzaki, J. Arai, T. Nakakawaji and Y. Ito, *Langmuir*, accepted for publication.
11. M. Arevalo, Masters Thesis, University of Oklahoma, (1994).
12. W.H. Waddell, J. O'Haver, L.R. Evans and J.H. Harwell, *J. Appl. Polym. Sci.*, **55**, 1627 (1995).
13. H. Chen, Masters Thesis, University of Oklahoma, 1992.
14. K. Esumi, N. Watanabe and K. Meguro, *Langmuir*, **7**, 1775 (1991).
15. D.T. Glatzhofer, G. Cho, C. Lai, E.A. O'Rear and B.M. Fung, *Langmuir*, **9**, 2949 (1993).
16. J. Wu, J.H. Harwell, and E.A. O'Rear, *J. Phys. Chem.*, **91**, 623 (1987).
17. L. Riviello, private communication.

18. J.H. O'Haver, L.L. Lobban, J.H. Harwell and E.A. O'Rear, in Solubilization in Surfactant Aggregates, J.F. Scamehorn and S.D. Christian Eds., (New York: Marcel Dekker). Chapter 8.
19. E.P. Plueddemann, Silane Coupling Agents, (New York: Plenum Press), 1991.

COATINGS SYSTEM EVALUATION AND DEVELOPMENT

**Michael R. Van De Mark, Director
University of Missouri-Rolla Coatings Institute
Department of Chemistry**

**University of Missouri
142 Schrenk Hall
Rolla, Missouri 65401**

**Final Report for:
Summer Faculty Research Program
Oklahoma City ALC
Tinker AFB, Oklahoma 73145**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, DC
and
Oklahoma City Air Logistics Center**

August 1995

COATINGS SYSTEM EVALUATION AND DEVELOPMENT

Michael R. Van De Mark, Director

University of Missouri-Rolla Coatings Institute

Abstract

The total coatings system utilized by the Oklahoma City Air Logistics Center was evaluated and studied. The system included the paint stripping, aircraft washing, skin brightening, chromate conversion coating, priming and painting steps. The problems isolated included: the incomplete removal of the paint, primer and cellulosic debris in the paint stripping operation; the ability of the soap to remove the oil from the underside of the wings in the washing step; the inability of the current chromating solution to produce a significant degree of conversion coating; and the need to have abrasion during the cleaning and skin brightening steps. The problems were evaluated by FT-IR and the use of spectrophotometric reflectance analysis of the chromated surface and the use of torque adhesion measurements to correlate the findings.

COATING SYSTEMS EVALUATION AND OPTIMIZATION

Michael R. Van De Mark, Director, University of Missouri-Rolla Coatings Institute

Introduction

The Oklahoma Air Logistics Center services the KC-135, the E-3, the B-1 and the B-52 airframes. These airframes are brought in for major servicing which includes mechanical repairs, electronics and repainting. There are basically two types of paint refinishing operations which are done on these air frames. The first is simply a scuff sand, prime and repaint. The second is a chemical strip followed by surface treatment, prime and painting. Many adhesion problems have been noted on virtually all airframes as well as problems obtaining a consistent high quality chromate pretreatment step. The purpose of this research and report are to detail the procedure and indicate procedural changes which could rectify current problems observed. The work covers the entire coatings system from surface preparation to final coatings including the stripping of the airframe.

1. Paint Stripping

1.a The system: The paint stripping system used at Tinker AFB is a phenol/methylene chloride based system which is designed to remove epoxy primer and urethane top coats. We found that the system contained a cellulosic thickener and this fact was confirmed by the supplier of the stripper. The thickener helps to keep the stripper on the vertical surfaces without running. The stripper is applied to the airframe, allowed to dwell and then the paint is removed with abrasion. Once the paint has been removed, the surface is rinsed with water. The resultant surface requires sanding to remove small residues of paint visible to the eye. At this point the airframe is then modified and repaired, a process which may take months.

1.b The Findings: The surface of the metal after stripping was covered with what looks like water spots and stains as well as some oil from the aircraft. In subsequent steps in the painting process the surface was cleaned. The residue left after the stripping process will need to be removed prior to painting. A chemically stripped panel from a B-52 was analyzed by microscopic FT-IR. The analysis clearly showed residue of urethanes from the top coat; epoxies, from the primer; and cellulose, from the stripper. These findings imply that the stripping operation is leaving paint and primer residue as well as residue from the stripper, the cellulosic thickener.

1.c Recommendations: If the surface receives a final rinse with a high pressure water jet of approximately 1500 lbs/in², the stripper residue may be minimized. This modification to the process will not add any process time to the operation but should make the subsequent steps more efficient. This suggestion has been made to Donald Svejksky, OC-ALC/TIESM and he has agreed to implement a test to verify this on a small scale.

2. Surface Cleaning

2a. The method: The first step in the actual painting process is washing the aircraft. The airframe is sprayed with a soap solution. This soap is allowed to dwell for a few minutes and then can either be power scrubbed with air drive abrasive pads or hand washed with red Scotch-Brite pads. The latter wash system is much less abrasive. Once the surface has been cleaned, the soap is rinsed off with water and a water break test is performed. If the surface beads water or the water sheet breaks, the surface must be rewashed until it passes this test. The final rinse is monitored with pH paper to insure that all the alkaline cleaner has been removed. The current process for all chemically stripped

aircraft is to power scrub the exterior. For scuff sanded airframes the hand wash method is still used even if some areas have been chemically stripped.

Eldorado 608 is the current soap and is a low foam cleaning system. Eldorado 608 was analyzed and found to contain silicone oil. This oil was used by Eldorado as an antifoaming agent to prevent foam. The low foam was a requirement added by the ALC to prevent foam which had become a problem in the waste treatment facility. The presence of silicone oil in the soap was subsequently verified by the manufacturer. Silicone oils can reduce adhesion and cause surface defects in coatings. They can cause cratering and loss of adhesion. In this same light, many products used by the ALC contain silicone oil. These products include hand lotion and skin protection creams. Many personal care products contain silicone oils and should be avoided in painting areas.

The water used in all the processes is "tap" water. The possibility of surface contamination or chemical reactions to produce insoluble by-products through contaminants in the water were investigated. The water was submitted for analysis with the following results:

Coliform	ND
Calcium	50 ppm
Iron	ND
Magnesium	23 ppm
Silicon	12 ppm
Sulfate	ND
Fluoride	ND
Chloride	ND
Bromide	ND

ND is below detection limits

The typical amount of water per square foot of aluminum surface on a drained but wet panel is approximately 2.0 g/ft² as determined on brightened aluminum. If the water residue is in the form of silicates and carbonates, 0.46 mg/ft² would be deposited dry. This amount of inorganic residue corresponds to a thickness of approximately 23 Angstroms of film (2.3 nanometers). Therefore, the residue may play a minor role in adhesion problems but cycling of the surface between wet and dry may deposit layers which could pose significant adhesion problems. The chemicals which are mixed with this water are the soap, skin brightener and the chromate. The performance of these chemicals may also be slightly altered by these impurities in the water. If the performance of the surface treatment system is adequate after the incorporation of the proposed changes, the use of tap water may be allowable and most economically expeditious.

2b. Questions and Evaluation: Three questions were raised regarding this step in the process. The first was how effective is the soap. The second was whether the abrasive cleaning was necessary. The third was whether the cleaning of the scuff sanded airframes was adequate and what was the differences in adhesion between these cleaning systems.

2b.1 The first question was evaluated through two sets of tests. The first was to place 0.4 g of aircraft engine oil into 60 ml of a 25% solution of different soaps to determine their effectiveness at the emulsification of the oils. Each tube was prepared and shaken vigorously several times and the rate of separation followed. The current soap, Eldorado 608, was a complete failure at the emulsification as was the blue soap based on MIL-C-87936. A new soap conforming to MIL-C-87937A (Eldorado 660) did a much better job of emulsification but separated after approximately 15 minutes. Dawn dish soap was tested as a comparison and it showed better emulsification than did any of

the other candidates. It is important to note that Dawn is partially ionic and cannot be used for airframe cleaning.

2b.2 The second question is to define the need for abrasive cleaning. To this end B-52 panels were washed with conventional red Scotch-Brite pads with no abrasion other than light movement of the pads and with scrubbing similar to the use of an abrasive pad to clean a pot with stuck on food. In fact the residual cellulosic thickener would be very similar to mashed potatoes which had been dried on for six months. After washing, the panels were brightened with the skin brightener, one without abrasion and the other with mild abrasion. They were both chromated with Allied Kelite 14-2 at 2oz/gal. The panel with abrasion was golden and uniform while the panel without the abrasion was mottled and had dark brown stained on the surface. Wiping the abraded panel with a white tissue resulted in a virtually clean tissue while the non-abraded panel produced a very dark brown patch on the tissue. The results clearly show that the abrasion is needed to remove the organic debris from the surface. In the past when the dark brown stains appeared on the aircraft, the skin was said to be "burnt" from over chromate treatment. This, however, was not likely the case but was due to an improperly cleaned surface with the organics being oxidized and causing the dark brown stain.

I did witness a B-52 being scuff sanded and cleaned. The bomb bay doors had been chemically stripped and then washed without abrasion. The doors were then skin brightened and chromated. The results were a very brown mottled chromate stain with the dark brown material having no adhesion. The stain rubbed off on a tissue as did the laboratory experiment. In the past, to limit this, time restrictions have been placed on the chromate exposure to avoid this problem erroneously. This process results in a very lightly chromated aircraft

with little corrosion protection. The organic debris on the surface is still there and adhesion will suffer. This material must be removed before chromate is applied.

2b.3 The third question relates to scuff sanded aircraft. The B-52 and B-1 are both currently scuff sanded airframes and are well known to have oil on their skins. This problem was evaluated through adhesion tests of scuff sanded B-52 skin contaminated with engine oil. A set of panels of B-52 skin removed from two different aircraft were used. The two skins are designated set A and set B since the two paint systems were different in age, possibly vendor and environmental exposure. Therefore, the two sets can not be compared to each other but panels within a set can be compared. A set of three panels were prepared each approximately 10" x10" for each test. All were wiped with an MEK dampened tissue to remove superficial contamination and then each of the panels was sanded with 240 grit sand paper. After sanding the panels were wiped with an MEK dampened tissue to remove the particulates. The panels were then contaminated with 0.5 grams of used jet engine oil and baked at 50°C for 24 hours to deeply embed the oil. After cooling to room temperature the panels were washed. Set A was washed with the oil side up. Three soaps were used at a 3:1 dilution; Eldorado 608, Eldorado 660 and MIL-C-87936. Each panel was first wet with 25 ml of the soap solution and gently spread. After a four minute soak the surface was agitated with a red Scotch-Brite pad and then 25 ml more of the soap was added and spread. After a second four minute dwell, the soap was agitated a second time and two minute dwell before rinsing with tap water. The panels were then allowed to dry for 1 hour at room temperature and 1 hour in an oven at 50°C before adhesion tests were performed. The results of the adhesion tests are given in Table I.

Set B was washed with the oil side down to emulate the under side of the wing and belly of the plane. Three soaps were used at a 3:1 dilution; Eldorado

608, Eldorado 660 and MIL-C-87936. Each panel was first wet with 25 ml of the soap solution using the Scotch-Brite pad and gently spread. After a four minute soak the surface was agitated with a red Scotch-Brite pad and then 25 more ml of the soap was added and spread. Unlike the oil side up panels, the soap solution dripped off and there was difficulty doing the test quantitatively. The panels were held by hand using handles glued on the back side of the panels. All panels were treated as reproducibly as possible. After a second four minute dwell, the soap was agitated a second time with a two minute dwell before rinsing with tap water. Upon inspection, a rainbow oil slick appeared on the wet panels washed with the Eldorado 608 and MIL-C-87936 soaps. The panels were then allowed to dry for 1 hour at room temperature and 1 hour in an oven at 50°C before adhesion tests were performed. The results of the adhesion tests are given in Table I. All the panels in this set had small oil stains on the perimeter of the panels indicating that the soaps were not removing all the oil from the surface and that a second washing would be necessary if all the oil were to be removed.

TABLE 1
Adhesion to B-52 Scuff Sanded
"Set A" "Set B"

<u>Soap</u>	<u>Wash</u>	<u>Adhesion</u> <u>lb/in</u>	<u>Wash</u>	<u>Adhesion</u> <u>lb/in</u>
Eldorado 660	Horizontal	3350	Inverted	4690
Eldorado 608	"	3610	"	4200
Mil-C-87936	"	3540	"	4272
STD	"	3050	"	4472

All values are the average of five tests.

Set A clearly shows that all three soaps do an adequate job of removing the oil. The current soap was slightly better than the others and the Eldorado

660 the poorest, however, their differences are very small. In Set B the trend is reversed with Eldorado 660 being the best. If one analyzes the data of this experiment in light of the soaps ability to emulsify the oil, it is not surprising that the Eldorado 660 was the best on the underside washing since it is the best at emulsification. Gravity acts to separate the oil in water system, since water is more dense than oil it goes down and the oil rises. On the underside of the wings the oil rises to the surface of the plane. However, on the top of the wing the oil will rise away from the surface of the plane. Thus, the need for a soap with a high emulsification capacity will be needed to remove the oil from the belly and under side of the wings and keep it suspended until it can be rinsed off. These emulsification tests must be run with both hydraulic fluid and engine lubricant use in the aircraft and not "internal combustion engine oil".

The Eldorado 660 soap was tested on a scuff sanded B-52 on the underside of one wing and the engine cowlings as a follow up experiment. The right wing, with respect to the pilot, received the Eldorado 660 soap and the left wing the old Eldorado 608. Workers found the new soap to be effective and that its slight odor was no problem. One worker said "the new soap made the floor more slippery on the left side of the plane", however, that side of the plane used the old 608 soap. Therefore, no problems were noted except that the new soap requires a few more minutes to rinse off. The flaps, which were new aluminum, were subsequently skin brightened and chromated. The 660 wing produced a more uniform chromated surface than did the 608, however, neither were adequate. Since hand cleaning was used, the new aluminum did not receive the abrasion needed to remove the surface film which prevents a good take of the chromate solution.

2.b.3.1 Adhesion Test: The adhesion test used in these studies was developed at Tinker AFB. The technique is a modification of the Patti Adhesion Test. A 0.5

inch diameter sand blasted disk is affixed to the surface with a very strong adhesive, MA300 Acrylic Adhesive from Plexus, 30 Endicott Street, Danvers, MA 01923. The disk is threaded on one end which allows a nut to be threaded on and subsequently torqued off. The torque needed to break the disk from the surface is related to the adhesive strength. A Mitutoyo Datatorq NO 983-201 digital torque wrench with 15 to 250 inlb range was used to determine the torque. This technique is more independent of metal thickness than the conventional Patti test. All adhesive strength measurements presented in this report were determined with this technique. The disks were allowed to cure for 24 hours before the torque measurements were made.

2.c Recommendations:

The current soap does not perform well on the underside and is not qualified. Eldorado 608 should be discontinued and Eldorado 660 should be used. Two new, approved soaps meeting military specification are available and should be evaluated in comparison to Eldorado 660 to determine which soap has the highest oil capacity. Adhesion tests should then be performed to verify that they will perform to the best level.

In addition to the soap change, it is recommended that all paint hanger personnel avoid the use of personal care products that contain: silicone oil, dimethicon, cyclomethicon or silicone on the ingredient label. These can be hand lotions, deodorants, cosmetics, hair conditioners, and related products. In most cases there are very good alternatives to these products. Care should be made not to use them during work or just before work since transfer to the aircraft from their hands or clothing would be more probable.

The surface of all new and chemically stripped areas on all airframes should be power washed and power scrubbed during the soap wash and skin brightening step to insure adequate chromate conversion coating.

3 Skin Brightening

3a. Method: The current skin brightener is Skin-Bright 555 from Omega and is to be used as a 1:1 dilution from concentrate. The dwell time on the surface is less than 10 minutes and is adhered to well by the paint hanger personnel. On chemical stripped aircraft, the brightener is applied to the surface and buffed with an air drive with green abrasive pads. On the surfaces of scuff sanded aircraft which are to be chromated, the brightener is applied and only hand rubbed with Red Scotch Brite pads resulting in very little abrasion and a mottled burnt appearance.

3b Questions: The obvious question is the need for abrasion and is the brightener doing its job.

3c Results: The experiments done in 2b.2 answer the questions above. The need for abrasion is obvious and without the skin brightener the surface would still contain the stains of the organic residue which neither the stripping nor the washing process were able to remove. It is well known that many polymers adsorb onto the surface of metals and their oxides. These adsorbed polymers are very tenacious and are virtually impossible to remove with solvents. The skin brightener actually removes the top layer of metal and its oxide but not where the organic adsorbed debris is. The result is to have raised areas protected by the adsorbed polymer and deeper etched areas analogous to a photo resist etch used in printed circuits. The brightening process exposes the native aluminum which the chromate solution needs to react with to form the passive layer which protects the aircraft from corrosion.

3d Conclusions: The skin brightener is obviously doing its job since tests show that it removes metal in accordance with its mission. The need to abrade the surface to remove the bulk of the organic debris is also necessary as can be

seen from the B-52 bomb bay doors which were not adequately surface prepared due to no power scrubbing and showed the undesirable "burnt" mottled appearance. Therefore as previously described, power scrubbing and abrasion during the washing process and skin brightening is necessary on all new and chemically stripped areas.

4. Chromate

4a. Method: The current chromate treatment is Alumigold at 1.25 oz/gal with less than 2 minutes surface dwell time. Problems in the past include: no gold color development, particles on aircraft, and "burning" of the skin.

4b. Findings

4b.1 Solid Impurities: The Alumigold was prepared as in production by dissolving 1.25 oz of the solid in water. The solution was warmed as advised by the manufacturer, but this was not done in the plant. In an analogous experiment a solution was prepared without heating with identical results as the heated material. The warmed solution was stored 24 hours and a white precipitate was filtered. The white precipitate contained a very small but distinctly different black precipitate. The black particles were mechanically separated and both materials were characterized by X-ray analysis. The results indicate that the white material is a silicate and the black a combination of barium, iron, chrome, and copper. The black was of minimal amount but the silica was substantial. The material can deposit on the aircraft during conversion coating and cause uneven chromating. This unevenness was noted on laboratory panels and was noted on actual aircraft.

4b.2 Color: The color of laboratory panels has often been used as a judgment of uniformity and degree of chromate "take" or development. The human eye is not quantitative but is sensitive to the color changes. Color is also very dependent

upon the person, the light source and the object. A better way to determine color in a quantitative way is through a reflectance spectrophotometer often called a color computer. This instrument is similar to the color measuring system in local paint stores used to match paint color. The device is a visible spectrophotometer which measures the reflected light at 400-700nm. Thus the color of panels can be measured through the use of this device.

We utilized the Minolta CM-2002 hand held color spectrophotometer with its Spectra Match software. The spectrophotometer has a measuring eye of approximately one centimeter in diameter. This small area will be susceptible to chromate variation on a given panel. Typical variation can be significant due to dwell time, chromate solution film thickness, local temperature, surface roughness, surface contaminants, and chromate concentration gradients due to the dilution from the water on the surface before chromating. Therefore to get an average of the surface at least 5 readings were made on all 3"x6" panels to gain a more representative color average. Even this average only represents approximately 4% of the surface. Edges should be avoided due to a number of edge effects including surface tension differences, local metal stress and possible contaminants migrating from the edge. The measurements reported here were taken two in the upper portion, two in the lower section and one in the middle. The color reported here is the average of that data.

Figure 1 shows the spectral reflectance from 400 to 700 nm for MRV1-7 and EDW which is a panel which had been chromated six months earlier in the laboratory with the Alumigold according to established procedures. These panels were cleaned with MEK and then with the Eldorado 608 soap followed by rinsing with tap water. The panels were then skin brightened for 10 minutes followed by a timed rise. Here MRV1 to 3 (5, 10, 15 min. respectively) are representative of tap water and MRV 4 to 7 (15, 10, 5 and 0.5 min respectively)

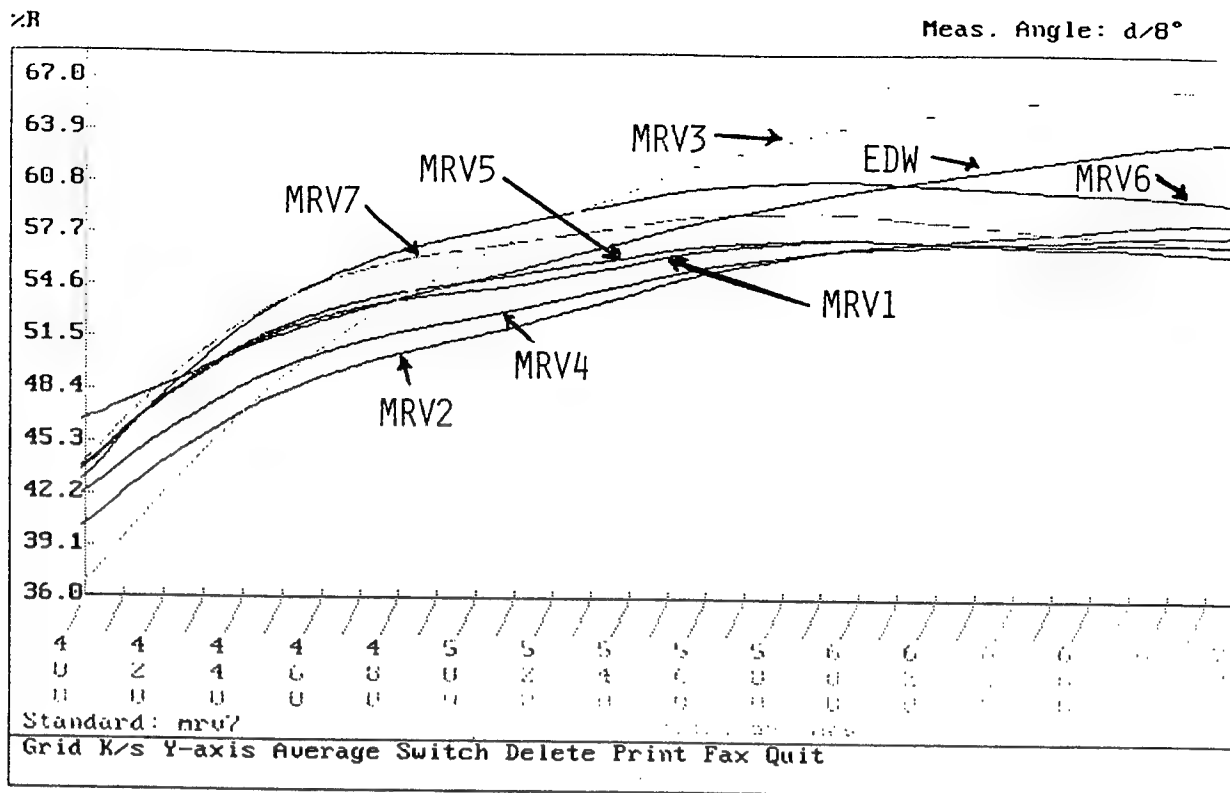


Figure 1. Chromate color development with tap and distilled water.

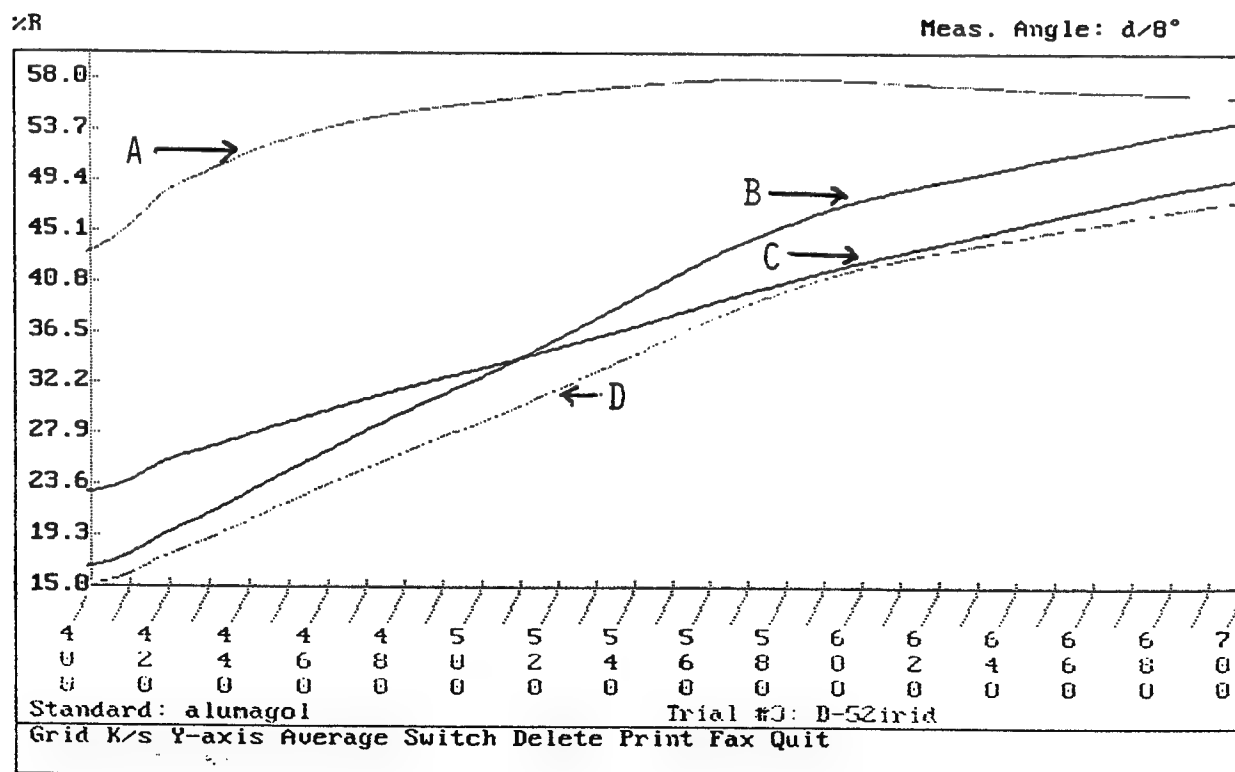


Figure 2. Chromate color development with A. Current Alumigold B. Iridite 14-2, C. "Burnt" B-52 with Iridite 14-2, D. B-52 properly cleaned/Iridite 14-2.

are using distilled water to rinse the skin brightener off. The variation is 5, 10 and 15 minute rinse time and MRV7 was a distilled water rinse of 30 seconds to establish a near zero point. All other samples were measured after a 24 hour drying period at 70°F and 50% relative humidity. It is clear that the major areas of differentiation is at 400 and 700 nm. Therefore all further data will discuss the values at these two points. The general shape of the curve is obviously important since water has a pronounced effect to linearize the curve and produce a browner color with a less distinct spectral signature. It is clear that on the surface chrome three and chrome six species are produced.

Figure 2 shows two conversion coatings applied to new aluminum cleaned with modest abrasion as with vigorous pot scrubbing. One is with the current chromate solution, A and the other with Iridite 14-2, B. Also shown is a chemical stripped B-52 cleaned by our current light abrasion, C showing the "burnt" color, and with the more vigorous pot scrubbing approach, D which is a clean gold color. The curve A shows little color development and thus relatively poor corrosion protection. Curve B illustrates a good chromate take and very good color development. Curve C shows an example of the "burnt" color development on chemical stripped aircraft. Here little detail is noted in the reflectance curve which is typical of the organic debris left on the surface being oxidized and stained by the chromate. Curve D is very similar to curve B but a little lower in reflectance. The drop in reflectance is due to the rougher surface and subsequent scattering of light and since there is more surface area, more light is absorbed by the increased chromed surface.

5 Adhesion Measurements: Donald Svejksky and Andrew Marshall at OC-ALC have developed an adhesion test utilizing an aluminum tab glued to the surface and measuring the torque needed to break the tab from the surface. The new torque test produces adhesion strengths which are typically about twice

that of the Patti test. This test is still undergoing development but does allow correlation of the adhesive strength of a coating or chromate without the need to have half inch thick aluminum as would be required with the Patti tester to obtain valid comparative data. All adhesive strength data reported here were performed using this procedure. The adhesive currently being used has an extremely short use life once mixed. We are currently seeking an alternative adhesive for laboratory use in this test.

TABLE 2
Adhesion of Chromate

Time hrs	A	B	C	D	E	F	G	Average
0.5	3320	1290	6090	3650	4820	-	-	3810 +/- 1800
1.0	3464	4120	5470	5820	2130	5240	-	4370 +/- 1410
2.0	2280	3250	2880	4530	5670	5620	-	4040 +/- 1450
4.5	4220	3220	5400	5370	4220	3720	-	4360 +/- 780
8	5550	6170	4130	4370	5770	5280	-	5210 +/- 800
17	2540	2090	4610	4110	5790	4880	-	4000 +/- 1420
25	3000	2870	5300	4390	4920	5880	-	4390 +/- 1230
49	3800	1440	4970	5060	5110	3700	-	4010 +/- 1410
72.3	3440	4120	4420	5640	4910	3630	4390	4360 +/- 750

The variable not defined was how long after chromate treatment does the surface have to dry before application of the primer. Table 2 shows data collected from three chromated panels of new aluminum prepared by the more abrasive route. The chromate treatment was with 2 oz/gal Iridite 14-2 for two minutes. The results are for dry times of the chromate from the moment the rinse was complete to the time the tab was glued on the chromated surface. The

letter designation refers to separate disks glued on the surface and the values of adhesion are reported in lbs/in². The data shows that after approximately four hours of drying the surface is at full adhesive strength. Examination of the mode of failure gives an analogous picture in that the failure occurs more with the adhesive than the chromate after 4 hours. The test data scatter was in part due to the glue being very difficult to work with to get consistent mixing. The age of the glue in the tubes can also vary the pot life when mixed and its subsequent holding strength. This problem must be addressed to verify the findings.

However, the test data does indicate that as long as new contaminants do not fall on the surface, the adhesive strength of the chromate will remain excellent for the full 3 days. In production the chromate is usually allowed 4 hours to dry since this is usually at shift change or lunch and masking must be done prior to priming.

6 Conclusions

The following are a list of findings and recommendations.

1. The final rinse in the paint stripping operation should be done with a high pressure water jet, 1500-3000 lbs/in² to remove the cellulosic thickener and residual paint.
2. Use Eldorado 660 cleaner in place of the Eldorado 608 currently being used. The additional soaps which meet the mil-spec should also be investigated, after switching to Eldorado 660, for oil emulsification and ability to remove oil from old paint in normal and inverted configuration.
3. Recommend that all employees avoid silicone containing personal care products before and during work.
4. Do abrasive cleaning of the new and chemically stripped aluminum surfaces to remove small amounts of residual organic debris.
5. Do abrasive scrub during the skin brightening step on all new and chemically stripped aluminum.
6. Switch from the current Alumigold chromate to Iridite 14-2 which produces a better chromated surface.
7. The chromating solution must be filtered to avoid the particulate ending up on the aircraft.
8. Remove the two minute dwell limit on the chromate treatment and use a color standard. The dwell time should be between limits of possibly 1.5 and 4.0 minutes. This time window must be worked out.
8. Establish a dry time before priming of 4 hours minimum if the relative humidity is below 75% and the temperature is over 70°F. The window of acceptability must be verified.
9. Research must be performed to optimize the chromate step. The concentration and application dwell time must be studied to find the optimum

with respect to corrosion protection and adhesion. We are proposing a follow up research program which will address this issue.

Acknowledgments

I wish to acknowledge the help and collaboration of Ed Weston and for the help of Donald Svejksky and Andrew Marshall on the adhesion studies.

**Substitute Cleaning Agents for CFC-113 in Oil and Grease
Removal From Airplane Parts**

**Marvin Fleischman
Professor
Chemical Engineering Department**

**University of Louisville
Louisville, KY**

**Final Report for:
Summer Research Program
Ogden Air Logistic Center**

**Sponsored by:
Air Force Office of Scientific Research
Bolling Air Force Base, Washington DC**

and

Ogden Air Logistic Center

October 1995

**SUBSTITUTE CLEANING AGENTS FOR CFC113 IN OIL AND GREASE
REMOVAL FROM AIRPLANE PARTS**

Marvin Fleischman
Professor
Department of Chemical Engineering
University of Louisville

Abstract

Organic and aqueous based materials were screened as potential replacements for CFC-113 in removing oil and grease during maintenance and repair operations of airplane parts, as per various technical orders (T.O.). An outside contractor will validate the substitute materials/processes. CFC-113 a chlorofluorocarbon, is an ozone depleting substance (ODS) whose manufacture is now banned under the Montreal Protocol. The comparative performance of each candidate cleaning agent with respect to CFC 113 was determined by flushing or ultrasonic cleaning of contaminated steel coupons based upon: qualitative solubility tests; percent weight removal of oil, grease and oil/grease mixture; Auger spectroscopy for surface analysis; and visual inspection. The screening tests showed the following candidates to be less effective than CFC 113: Dry Wipes (no solvent), Deionized Water, Alkaline Cleaner (EZE 445), OS-20 (methyl polysiloxane), OS-30, n-Methyl Pyrollodine, Isopropanol, Citrikleen HD Spray & Wipe, & Penair HD-1 (terpene). The following candidates were almost, as, or possibly more effective than CFC 113: Aliphatic Naptha, AK225 (hydrochlorofluorocarbon), and DS 108 Wipe Solvent (petroleum solvent mixture).

Introduction

Parts come in due to failure. It is generally unknown if the part is dirty, but it is cleaned anyhow for protection or routine maintenance. Various technical orders (T.O.) reference use of 1,1,1-trichloroethane (TCA) and/or CFC113. But, CFC113 is primarily used. CFC113 & TCA are ODCs. CFCs are no longer produced, & TCA production is being phased out. The Air Force is thus committed to finding environmentally compatible substitutes. In addition to ozone depletion, consideration must be given to global warming potential (GWP), volatile organic compound (VOC) content, flammability, & toxicity of alternatives.

The processes addressed by the T.O.s include vapor degreasing, ultrasonic mixing, flushing, & hand wipes. Contaminants include light dirt/dust, hydraulic oil (MIL-H-5606E,F,G), & grease (VV-P-236). Substrates are primarily stainless steel, steel & aluminum. Some specific T.O. requirements & cleanliness criteria include: Vapor degrease parts (visually clean); Immerse free screwjack assembly (must free-wheel without hesitation, a functional cleanliness measure); Rinse/flush metallic parts (visually clean, enable bonding of ablative material); Vapor degrease & ultrasonically clean (visually clean, enable bonding); Flush hydraulic oil from arm channels (visually clean).

TIELM was tasked with identifying & screening non ODS candidate materials & processes to eliminate CFC113 in maintenance & repair operations. Various solvents were to be lab tested for oil & grease removal. An outside contractor will validate the substitute materials/processes. Partial results of the screening tests are presented in this report.

Selection of Candidate Materials

Candidate materials were identified by both TIELM & the contractor. Selection of potential substitute organic & aqueous based cleaners was based on experience, chemical manufacturer's materials safety data & technical data sheets, & direct contact. To a very limited extent, solvent selection was partially based

on literature review & theory. Thus some candidates were selected (or eliminated) based on "like dissolves like", i.e., polar solvents readily dissolve polar contaminants, & vice versa. A candidate could thus be identified by comparing structure, functional groups, & molecular weight of cleaning agent & contaminant.

Many potential candidate materials were identified, but actual tests were limited to those readily available. Others were eliminated for anticipated poor cleaning ability, e.g.: Ionox - Polar solvent would probably not work with non-polar contaminants; L-12862-Contains perfluorocarbon with limited solvent power. The hydraulic oil contaminant was a water insoluble petroleum based fluid. The petrolatum grease was semi-solid opaque, & oil like, & primarily mineral oil (mixture of high molecular weight paraffins).

Test Procedure

Comparative performance of each candidate material with respect to CFC113 was determined by ultrasonic cleaning of contaminated steel coupons at ambient temperature for 5 minutes, followed by weighing, microscopic surface analysis (Auger Microprobe), & visual inspection. The extent of contaminant removal (% weight) was determined. Visual & qualitative parameters included: surface residue/corrosion, cleaning ease, & drying time. Aqueous candidates were formulated by deionized water(DI) dilution. At times a wash bottle flush was used instead of ultrasonics, & dry wiping sometimes followed ultrasonics (as the agent should be used, & for slow driers). A DI rinse was used when specified manufacturer, sometimes followed by laboratory airblowing for drying.

CFC113 was the reference solvent because of specified use in the T.O.s. Ultrasonic cleaning was selected for convenience, and because it is one of the stronger and common cleaning methods. Tests were at ambient temperature, typical of ultrasonic cleaning, though the bath does heat up. Mixing was 5 minutes, although 15 minutes is typical. The longer time might completely

remove the contaminants, indicating all candidates to be equally effective. Cleaning with flammable agents was done in a beaker placed in a special cover to avoid direct contact of solvents with the ultrasonic hardware.

In the tests, coupons were generally processed as follows:

- a) Ultrasonically cleaned at ambient temperature with CFC-113
- b) Weighed
- c) Contaminated with oil, grease, or oil/grease mixture & reweighed
- e) Ultrasonically cleaned at ambient temperature with candidate (or with a wash bottle flush)
- f) Air dried & reweighed (DI water rinse & air blown for some materials, patted dry or dry wipe for others prior to reweighing)
- g) Visually inspected for residue or other surface defects
- i) Auger surface analysis

Oil was applied by a syringe & grease by the wooden end of a cotton swab. For oil/grease mixture (actual field contaminant), grease was first applied & oil added from the cotton end of a swab, also used to mix the oil & grease. Oil & grease were mixed to simulate interactions which might occur during field operation. For some tests contaminant was baked onto coupons because parts operate at 180F & the oil is gummy. In addition to cleaning tests, some qualitative solubility tests were done.

The true surface, which affects bonding & lubrication, is the first few atomic layers. Atomic concentrations of various elements, e.g., carbon, can be determined by Auger Spectroscopy. Information on atomic concentration at various penetration depths (contaminated layer thickness) can be obtained by sputtering away the surface with an argon ion gun & re-analyzing the new surface with the electron beam probe. A typical analysis is shown below. Concentrations at lower depths don't provide accurate information because of contaminant redeposition on the clean surface resulting from sputtering. Craters will make Auger photos difficult to interpret due to shadowing.

For screening candidate materials, only the carbon

concentration (peak) is reported because the contaminants were primarily hydrocarbons as were many candidate cleaners. Thus, the carbon peak best indicates residual cleaning agent & relative contaminant removal. Electron microscope photos of the surfaces (Photos 1-24) also provide visual information on impact & effectiveness of candidates.

Auger Analysis

Atomic Concentration(%) Summary Table

File Name rea Cycle	C1	Ca1	01	Fe3	S1	P1	Sb1	Cl1	N1	Al1	Si1	Mo1
5010.2												
1 1	81.94	2.44	9.79	5.83
5010.3												
1 1	38.43	2.70	28.62	30.26
5010.4												
1 1	28.28	1.17	30.89	39.65
5010.5												
1 1	16.31	1.14	21.07	61.48
5010.6												
1 1	11.02	1.74	13.75	73.48

#1 Virgin

900 Å total

Baselines (fingerprints) were established by Auger analyses on virgin coupons, virgin coupons cleaned with 1,1,1-TCA & CFC-113, & virgin coupons cleaned with CFC to which oil, grease, & oil/grease, resp., were added. The experimental results are summarized in Table 1.

Experimental Results & Discussion

Baked oil appeared to be as fluid as unbaked oil rather than gummy. Comparing peaks & penetration for the baseline coupons given in Table 3, indicates that grease may penetrate further than oil. Comparing density of dark areas in Photos 3 & 4, also indicates that grease is a stronger contaminant. Carbon concentration should be directly proportional to the weight of applied contaminant. A better measure of relative contamination between oil & grease might be to compare them on the basis of peak ht./contaminant weight. The candidate materials are now compared to CFC113.

CFC 113 Cleaning: In testing solubility, oil dissolved & grease dispersed into smaller particles. On a weight basis, flushing completely removed oil; however, grease was not as easily removed. This test & subsequent ease of oil removal with other

agents indicates that oil is easy to remove mechanically, probably due to fluidity & relatively lower adhesion to metal. Flushing could be tested on baked coupons. Auger analyses is a better indicator of effective cleaning than weight removal.

Ultrasonics completely removed all contaminants macroscopically. Comparison of carbon peaks to grease baseline showed a significant decrease. Comparing Photos 7 & 4 shows a cleaner surface after CFC cleaning (original photos show differences much clearer). Likewise oil & oil/grease, un-baked & baked, peaks after cleaning were significantly reduced compared to baselines. Comparison of Photos 6, 7, 8, & 9 show a cleaner surface after oil removal than for grease & oil/grease. Comparing Photos 8 and 9 (shows residue) indicates that baking could make oil/grease cleaning more difficult. This is further supported by the lower surface peak for the un-baked coupon, despite the significantly higher weight of applied contaminant for the un-baked sample (0.0428 vs. 0.00238 gms).

On virgin coupons, grease penetrated further than oil; on CFC cleaned contaminated coupons, oil and grease peaks and penetration were similar. The oil's fluidity might allow deeper penetration into surface crevices. Surface peaks after cleaning unbaked oil/grease were about the same as for cleaning oil alone. However the concentration was lower for oil at the same depth suggesting that oil fluidity might be reduced by mixing with grease. Some carbon at various depths could be from residual solvent as well as contaminant. For CFC, this could be checked by Cl concentrations at various depths.

Dry Wipe (no solvent): 100% & 99% removal of oil and grease resp., was achieved, indicating that macroscopic contaminant removal is not difficult. However, in both cases surface residue was visible, & Photo 10 indicates wiping to be much less effective than CFC for oil removal. Likewise, the oil removal peak is significantly higher for the wipe both at and below the surface. In fact, carbon concentrations after oil removal by wiping weren't that much lower than the baseline for oil.

Deionized(DI) Water: Water was not expected to be effective

because of the contaminants' aliphatic nature. Also, water would take longer to dry than CFC. However, while CFC was the acceptable cleaning control, water would represent the worst case. Grease removal by ultrasonic mixing was negligible & the coupon was not clean. However, oil removal was 97%, again indicating ease of macroscopic oil removal. However, the surface peak was much higher than for CFC & carbon penetration carbon was much higher for water. The water cleaned coupon was blown dry with lab air, which could have contributed to contamination. It was later discovered that the air wasn't clean. The need for air blowing points out that aqueous cleaners require drying by air blowing or a vacuum oven, whereas CFC cleaning requires only air drying.

Alkaline Cleaner: Oil & grease removal were both 100%(wt). The coupon was tarnished after both oil & grease removal, & rust spots were evident on a CFC cleaned uncontaminated specimen after ultrasonic mixing with the alkaline cleaner. The cleaner may not contain sufficient or adequate rust preventative. For grease removal, the peaks & penetration were greater than for CFC. Photo 12 also shows that alkaline cleaner was not as good. For oil removal however, peaks & penetration were less than for CFC, indicating that alkaline cleaner performed better except for the drying requirement & rusting. However, comparison of Photo 13 indicates a cleaner surface than for CFC. Oil is expected to be easier to remove because of the grease's higher molecular weight.

OS-20: This material (also OS-30) is a volatile linear methyl siloxane, non-VOC, & low GWP. It has a high solubility parameter and Kauri-Butanol value, but is flammable. Flushing removed 100%(wt) of the oil & 56% of the grease. Grease removal in the ultrasonic was 98-99%, but some grease seemed to remain & there were surface blemishes. Auger peaks & penetration were higher than for CFC & Photo 14 shows higher surface contamination. Oil removal in the ultrasonic was 100%, & the surface looked clean. Peaks were higher than for CFC; Photo 15 shows a dirtier surface.

OS-30: Flushing did not remove any grease, but 99-100% was removed by ultrasonics. Residual solvent dried slowly & had to be

patted dry. A blemish was visible. Peaks & penetration were higher than for CFC. Oil removal by ultrasonic was 100%, & peaks were higher than for CFC.

n-Methyl Pyrrolidine: This was tested because it is one of the more promising replacements for chlorinated solvents. In the solubility test, oil did not dissolve and grease was dispersed. Ultrasonic mixing gave a clear yellow solution indicating some grease solubility. Some oil dissolved with stirring.

Almost 100%(wt) grease removal was obtained. The coupon appeared clean prior to a DI rinse & air blow drying, but the dried coupon was tarnished. Peaks & penetration were significantly higher than for CFC. Photo 7 also indicates that nmp is not as effective as CFC. Oil removal was not run because of nmp's poor grease removal, which should eliminate it from further consideration. Actually nmp is used mostly to replace methylene chloride or methyl ethyl ketone, & should not be expected to be effective for oil and grease removal.

Aliphatic Naptha: Naptha was chosen based on "like dissolves like"; both oil & grease are aliphatic & non-polar as is naptha. However, flammability would be a concern & naptha probably could only be used under appropriate safety measures. Both oil & grease dissolved readily. Flushing yielded 100% oil removal with no residue, and 68-73% grease removal. Grease removal was less than for CFC flushing, & perhaps this test should be rerun.

Ultrasonics yielded 100% removal of all contaminants. There was a residue after unbaked & baked grease removal, & carbon peaks were greater than for CFC cleaning both at the surface & at 750A. Photo 17 also indicates that naptha is not as effective as CFC 113. No residue was visible after unbaked & baked oil removal. Again Auger peaks were higher than for CFC, but closer than the peaks for grease removal. Photo 18 also indicates that naptha was not as effective as CFC for oil removal. Peaks after removal of baked grease/oil were also higher, & there was a residue. However Photo 19 indicates a cleaner surface from naptha.

AK225-AES: (Hydrochlorofluorocarbon [HCFC] 225ca,cb). Boiling

point is slightly above CFC113, & HCFC comes as close to a drop in replacement for CFC113 as any substance currently identified. Thus, HCFC225 could possibly be used in existing equipment, e.g., vapor degreaser, until it's intended phaseout by 2030. HCFC225 ca/cb is an acceptable CFC113 substitute for solvent cleaning in electronics & precision cleaning, subject to the exposure limit of 25 ppm of the ca isomer.

Grease did not dissolve, but oil did readily. Flushing gave 100% oil removal. Grease removal was only 2.3%, but compared to the other solvents this seems far too low and perhaps this test should be rerun. In the ultrasonic, oil & grease removal were both 100% & the coupons appeared clean. Tests were not run on oil/grease. Carbon peaks were higher than for CFC both at the surface & at 750A for grease removal, & lower for oil removal. **DS-108 Wipe Solvent:** (Flash point ~115F, possibly flammable). On baked oil/grease removal, removal was 75% before wiping & 100% after. Auger peaks were about the same as for CFC, & by Photo 20 the surface may be cleaner than for CFC. Wiping was done because the cleaning agent evaporated slowly & the agent is intended to be used with a wipe. The coupon appeared to be clean prior to wiping, but residue remained after.

Isopropanol: Based on polarity considerations, isopropanol(ipa) might not be effective in removing paraffinic grease. Ipa is also flammable & might be difficult to use in a vapor degreaser. However, ipa was specified by the outside contractor because of general utility as a chlorinated solvent alternative. In the solubility test, oil dissolved but not grease. A flush yielded 56% removal of unbaked oil/grease mix. Ultrasonic cleaning of oil/grease yielded 93% removal. Auger peaks were slightly higher than for CFC, but penetration was much greater.

Citrikleen HD Spray & Wipe (semi-aqueous): This is a biodegradable limonene based terpene also containing fatty acid soap, diethylene glycol monobutyl ether, ethanolamine, nonionic surfactant, dipropylene glycol, diethanolamine, & pine oil. It is classified as a solvent emulsion, i.e., oils & greases may rise when left standing. It has a low evaporation rate & the cleaned

coupon was rinsed with water. Weighing tests for oil/grease removal got messed up. Carbon peaks & penetration were much higher than for CFC.

Penair HD-1: This is a water based biodegradable with D-limonene (25-30%). Ultrasonic cleaning followed by a DI rinse yielded 60% removal of unbaked oil/grease. A grease blob remained after cleaning & the surface appeared corroded (could be due to DI rinse). Auger peaks were much higher than for CFC & penetration was much greater. Photo 22 further indicates that Penair is less effective in removing oil/grease than CFC.

OCS-NFH2002 (Immersion cleaner): This was used incorrectly. Cleaning of oil/grease mixture was done with undiluted material, and it is typically used at a 1:8 water dilution for ultrasonic cleaning at 70-160F. Hence this test should be repeated at proper dilution. Used without dilution, 90%(wt) removal of oil/grease was achieved with ultrasonic cleaning. The precipitate that formed would not dissolve in CFC, ipa, or acetone, but did dissolve in naptha indicating that the precipitate was primarily paraffinic & probably resulted from grease.

Conclusions & Recommendations

Based upon the results of the screening tests, the following candidates are considered **less effective than CFC 113:**

Dry Wipes, Deionized Water, Alkaline Cleaner EZE 445 (may be more effective for oil removal), OS20, OS30, NMP, ipa, Citrikleen HD, & Penair HD-1. The following candidates were **almost, as or possibly more effective than CFC 113:** Aliphatic Naptha, AK225, & DS 108.

Further testing of these cleaning agents is recommended, perhaps adding water break tests, Fourier Transform Infrared (FTIR) analysis of extract after cleaning (peaks identify extracted compounds from contaminants & peak height indicates relative removal), and ESCA (a surface sensitive analysis technique also known as X-Ray Photoelectron Spectroscopy which can determine surface composition of elements and their chemical state).

Table 1
Experimental Results
Removal of Oil and Grease by Various Solvents (Cleaning Agents)

Specimens (to which contaminants were added): 4130 Steel (Carbon: 0.27-0.33%), ~2" x 0.5" x 1/16" thick, 3.3-3.6 gms. Specimen initially cleaned for 5 minutes with Freon 113 at ambient temperature in a Branson 2200 ultrasonic cleaner at 50/60 Hz unless stated otherwise prior to contaminant(s) to specimen.

Contaminants:

- + Oil (hydraulic, Brayco Micronic 756B, MIL-H-5606): Low viscosity fluid, dyed red, petroleum base, contains additives-viscosity/temperature coefficient improvers, oxidation inhibitors, anti-wear agent
- + Grease (Petrolatum, Technical, MIL-VV-P-236A): Solid

Analytical Instrumentation:

- + Mettler AE260 Delta Range Analytical Balance - accurate to four decimal places
- + Perkin Elmer Electronics 660 Scanning Auger Microprobe - Lanthanum boride electron beam probe, Argon ion gun (for sputtering surface)

Solvent Cleaning Agent & Type	Contaminant	Wt. of applied Contaminant (gms)	Removal of Contaminant by Solvent (% wt.)	Relative Auger Carbon Peak & Surface Penetration	Auger Photograph (Fig. #)	Comments
None-Virgin (Uncleaned) Specimen	None			81.94 @ surface 11.02 @ 900 Angstroms (Å)	1	
1,1,1 Trichloroethane wipe	None-Cleaning raw specimen			66.20 @ surface 11.82 @ 750 Å		
CFC(Freon) 113	None-Cleaning raw specimen (base line)			54.14 @ surface 7.81 @ 750 Å	2	
None	Oil (baked) on CFC cleaned virgin specimen-base line			98.50 @ surface 12.23 @ 2400 Å	3	Baking was done at 185F for ~ 20 hrs.
	Grease (baked) on CFC cleaned virgin specimen-base line			100 @ surface 76.88 @ 4800 Å	4	

		Oil & grease (baked) on CFC cleaned virgin specimen - base line	0.0026		94.25 @ surface 25.79 @ 2600A	5	
Visible residue @ 96% grease removal							
Chlorofluorocarbon (CFC 113) Flush w wash bottle	Oil	0.0084	100				
	Grease	0.0523 0.0190	11-12 73-96				
CFC 113 Ultrasonic	a) Oil	a) 0.0121	a) 93	a) 58.07 @ surface 8.39 @ 750A	a) 6	a) Visible residue, even after cleaned specimen flushed with CFC	
	b) Oil (baked)	b) 0.0065	b) 100				
	a) & b) Grease	a) 0.0376	a) 100	a) 16.94 @ surface 9.58 @ 750A	a) 7	a) Initial TCA clean: 15 min. with CFC 113 in ultrasonic b) CFC squirt prior to ultrasonic c) 180F, 65+ hrs. Residue appears same as for a)	
	c) Grease (baked)	b) 0.0271 c) 0.0167	b) 100 c) 100				
	a) Oil & grease	a) 0.0428	a) 100	a) 54.18 @ surface 8.81 @ 750A	a) 8		
	b) Oil & grease (baked)	b) 0.0238	b) 100	b) 62.95 @ surface 13.76 @ 750A	b) 9		
Beaker tests-oil dissolves in CFC; grease is dispersed & mixture is hazy after ultrasonic mixing							
None wipe only	Oil	0.0347	100	90.74 @ surface 17.25 @ 1200A	10	Heavy residue after cleaning	
	Grease	0.0675	99				
Deionized water	Oil	0.0091	97	100 @ surface	11	Light film Cleaned specimen air	

				17.17 @ 4350A	blown to dry. Water was cloudy, residual film				
Grease				0.0541	4.6	Air blown, did not look clean			
						Beaker test - oil & grease didn't dissolve; addition of alconox yielded cloudy mixture w oil & grease floating			
Alkaline (EZE 445) - MIL-C-85570: 1 part cleaner to 9 parts DI H ₂ O; pH 8.5				Grease	0.0239	100	94.48 @ surface 13.77 @ 1950A	12	DI rinse of cleaned part, followed by air blow. Cloudy mixture after both oil & grease removal, & tarnished specimen in both cases. CFC cleaned raw specimen showed rust spots after washing with solution in ultrasonic for 5 mins.
				Oil	0.0191	100	22.02 @ surface 5.80 @ 450A	13	
OS-20 (volatile methyl polysiloxanes) (Dow Corning) Flammable				Grease	a) 0.0208 (flush) b) 0.0589	a) 56 b) 98-99	b) 70.70 @ surface 12.19 @ 1200A	b) 14	b) Blenishes or residue
				Oil	0.0110	100	70.33 @ surface 11.54 @ 750A	15	Surface looked clean
									Beaker test - Oil dissolved; cloudy fluid w grease, grease particles settled out
OS-30 (volatile methyl polysiloxanes) (Dow Corning) Flammable				Grease	a) 0.0269 (flush) b) 0.0441	a) ~0 b) 99-100	b) 65.30 @ surface 12.21 @ 1200A		Slow drying, patted dry. Residue or blenish

n-methyl pyrrolidine (nmp) semi-aqueous	Oil	0.0080	100	70.04 @ surface 11.98 @ 450A	16	Cleaned sample patted dry • Specimen appeared clean. Cleaned specimen then patted dry, rinsed w DI H ₂ O, air blown, visible tarnish. • Grease particles floating in final wash. • Beaker tests - Grease disperses and gives mixture yellow color; 25 min. ultra-sonic gave clear yellow soln. w grease floating. Oil & grease added to nmp, ultrasonic for 15 mins.; floating oil layer formed.
	Grease	0.0743	99-100	97.05 @ surface 17.25 @ 2400A		
Aliphatic naphtha (20F flashpoint, 206F boiling pt.) Flammable	Grease	a) 0.0384 (flush) 0.0190 (flush)	a) 68 73	b) 61.91 @ surface 11.33 @ 750A	b) 17	a) No visible residue b) Visible residue c) Visible residue
		b) 0.042	b) 100			
		c) 0.0174 (baked, 180F, 60+ hrs.)	c) 100			
Oil		a) 0.0094 (flush) b) 0.0205	a) 100 98	b) 67.77 @ surface 12.32 @ 750A	b) 18	All samples look clean
		c) 0.0005 (baked)	c) 100			
Oil & grease (baked, 180F, ~20 hrs.)		0.0212	100	74.59 @ surface 12.63 @ 750A	19	Visible residue Beaker tests - Oil & grease readily dissolved

AK 225-ABS Hydrochlorofluoro- carbon (HCFC) Asahi Glass, Japan Global warming potential, Phaseout by 2030	Grease	a) 0.0396 (flush) b) No additional grease added after flush	a) 2.3 b) 100	b) 52.25 @ surface 9.61 @ 750A	Surface looked clean for both oil & grease removal Beaker test - oil dissolved, grease did not
DS-108 Wipe Solvent (mixture of petroleum solvents & other ingredients)- Dynamold Solvents	Oil & grease (baked, 180F, ~20 hrs.)	0.0040	a) 75 w/o wipe b) 100 w wipe	b) 61.36 @ surface 13.34 @ 750A	Solvent evaporates slowly; visible film prior to wiping; no film after wiping, but visible residue
Isopropanol Flammable, VOC	Oil & grease	a) 0.0187 b) 0.0117	a) 56 b) 93	b) 58.14 @ surface 10.14 @ 900A	Bicamish after ultrasonic Beaker test - Oil dissolves; grease doesn't
Citriklean HD Spray & wipe (limonene based terpene) - 4 parts DI water to 1 part cleaner-Biodegradable	Oil & grease	0.0251	60	91.16 @ surface 18.34 @ 3000A	• DI H ₂ O rinse after ultrasonic • Grease blob on cleaned specimen • Cleaned surface is tarnished (possibly from water rinse)
OCSS-NFH2002* (Immersion cleaner, 20% butyl cellosolve, glycol ether - O.C.S. mfg.	Oil & grease	0.0279	90	67.96 @ surface 45.41 @ 5100A	• Solvent evaporates slowly • After cleaning, cloudy mixture • Purple color & crystalline white precipitate after 2 days - ppt. dissolved in aliphatic naphtha

• This cleaner was used straight. Should have been used at a 1-2 to 1-8 dilution with water.

Auger Photographs (Baseline & contaminated samples after cleaning)



1. Virgin (untreated) specimen



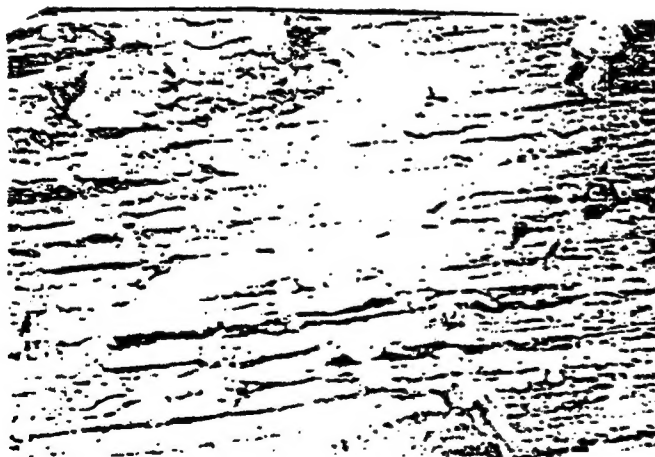
2. Virgin specimen cleaned with CFC 113



3. CFC cleaned virgin specimen, baked oil



4. CFC cleaned virgin specimen, baked grease



5. CFC cleaned virgin specimen, baked grease
& oil



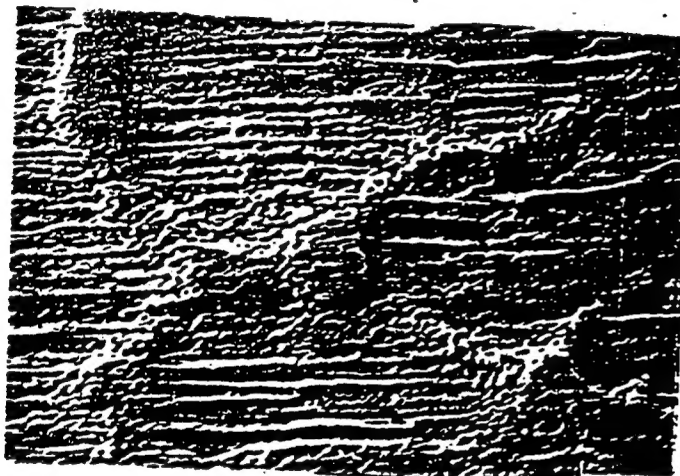
6. Oil contaminant on cleaned virgin specime.
CFC cleaning of contaminant-ultrasonic



7. Grease contaminant, CFC cleaned



8. Oil & grease, CFC cleaned



9. Baked oil & grease contaminant, CFC cleaned



10. Oil, wipe only (no solvent)



11. Hydraulic oil, deionized water



12. Grease, MIL-C-85570



13. Oil, MIL-C-85570



14. Grease, OS-20



15. Oil, OS-20



16. Grease, NMP



17. Grease, Naptha



18. Oil, Naptha



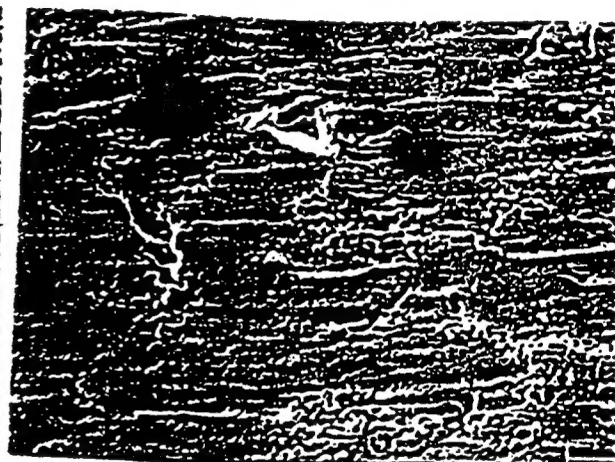
19. Baked oil & grease, Naptha



20. Oil & Grease, DS-108



21. Oil & Grease, Citrikleen



22. Oil & Grease, Penair HD



23. Oil & Grease, NFH2002